Polynomial-Time Verification of PCTL Properties of MDPs with Convex Uncertainties

Alberto Alessandro Angelo Puggelli
Wenchao Li
Alberto L. Sangiovanni-Vincentelli
Sanjit A. Seshia

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2013-24
http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-24.html

April 3, 2013

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
03 APR 2013		**00-00-2013 to 00-00-2013**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Polynomial-Time Verification of PCTL Properties of MDPs with Convex Uncertainties	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Berkeley,Electrical Engineering and Computer Sciences,Berkeley,CA,94720**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We address the problem of verifying Probabilistic Computation Tree Logic (PCTL) properties of Markov Decision Processes (MDPs) whose state transi- tion probabilities are only known to lie within uncertainty sets. We rst introduce the model of Convex-MDPs (CMDPs), i.e., MDPs with convex uncertainty sets. CMDPs generalize Interval-MDPs (IMDPs) by allowing also more expressive (convex) de- scriptions of uncertainty. Using results on strong duality for convex programs, we then present a PCTL veri cation algorithm for CMDPs, and prove that it runs in time polynomial in the size of a CMDP for a rich subclass of convex uncertainty models. This result allows us to lower the previously known algorithmic complexity upper bound for IMDPs from co-NP to PTIME. Using the proposed approach, we verify a consensus protocol and a dynamic con guration protocol for IPv4 addresses.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**33**	

1 Note to the Reader

This technical report is the extended version of the paper *"Polynomial-Time Verification of PCTL Properties of MDPs with Convex Uncertainties"* presented at the 25^{th} International Conference on Computer Aided Verification, CAV 2013. The report extends the conference submission with the following additional material, which was not added due to space limitations:

1. **Appendix A**: Results on convex optimization.
2. **Appendix B**: Alternative verification procedure based on Value Iteration.
3. **Appendix C**: Linear programming formulation to verify the property including the Until operator on the running example introduced in the paper.
4. **Appendix D**: Case study on the verification of the Dining Philosopher problem.

Polynomial-Time Verification of PCTL Properties of MDPs with Convex Uncertainties

Alberto Puggelli, Wenchao Li, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia

Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
{puggelli, wenchaol, alberto, sseshia}@eecs.berkeley.edu

Abstract. We address the problem of verifying Probabilistic Computation Tree Logic (PCTL) properties of Markov Decision Processes (MDPs) whose state transition probabilities are only known to lie within uncertainty sets. We first introduce the model of Convex-MDPs (CMDPs), i.e., MDPs with convex uncertainty sets. CMDPs generalize Interval-MDPs (IMDPs) by allowing also more expressive (convex) descriptions of uncertainty. Using results on strong duality for convex programs, we then present a PCTL verification algorithm for CMDPs, and prove that it runs in time polynomial in the size of a CMDP for a rich subclass of convex uncertainty models. This result allows us to lower the previously known algorithmic complexity upper bound for IMDPs from co-NP to PTIME. Using the proposed approach, we verify a consensus protocol and a dynamic configuration protocol for IPv4 addresses.

1 Introduction

Stochastic models like Discrete-Time Markov Chains (DTMCs) [1] and Markov Decision Processes (MDPs) [2] are used to formally represent systems that exhibit probabilistic behaviors. These systems need *quantitative* analysis [3] to answer questions such as "what is the probability that a request will be eventually served?". Properties of these systems can be expressed and analyzed using logics such as Probabilistic Computation Tree Logic (PCTL) [4] — a probabilistic logic derived from CTL — as well as techniques for probabilistic model checking [5]. These methods often rely on deriving a probabilistic model of the underlying process, hence the formal guarantees they provide are only as good as the estimated model. In a real setting, these estimations are affected by uncertainties due, for example, to measurement errors or approximation of the real system by mathematical models.

Interval-valued Discrete-Time Markov Chains (IDTMCs) have been introduced to capture modeling uncertainties [6]. IDTMCs are DTMC models where each transition probability lies within a compact range. Two semantic interpretations have been proposed for IDTMCs [7]: Uncertain Markov Chains (UMCs) and Interval Markov Decision Processes (IMDPs). An UMC is interpreted as a family of DTMCs, where each member is a DTMC whose transition probabilities lie within the interval range given in the UMC. In IMDPs, the uncertainty is resolved through non-determinism. Each time a state is visited, a transition distribution within the interval is adversarially picked, and a probabilistic step is taken accordingly. Thus, IMDPs model a non-deterministic choice made from a set of (possibly uncountably many) choices. In this paper we do not consider UMCs and focus on IMDPs.

An upper-bound on the complexity of model checking PCTL properties on IMDPs was previously shown to be co-NP [8]. This result relies on the construction of an equivalent MDP that encodes all behaviors of the IMDP. For each state in the new MDP, the set of transition probabilities is equal to the Basic Feasible Solutions (BFS) of the set of inequalities specifying the transition probabilities of the IMDP. Since the number of BFS is exponential

Table 1: Known Upper-Bound on the Complexity of PCTL Model Checking.

Model	DTMC [4]	IMDP [8]	IMDP/CMDP [ours]
Complexity	PTIME	co-NP	PTIME

in the number of states in the IMDP, the equivalent MDP can have size exponential in the size of the IMDP. In this paper, we describe a *polynomial-time algorithm* (in both size of the model and size of the formula) based on Convex Programming (CP) for the same fragment of PCTL considered in [7, 8] (the *Bounded Until* operator is disallowed). This shows that the problem is in the complexity class PTIME. With *Bounded Until*, the time complexity of our algorithm only increases to pseudo-polynomial in the maximum integer time bound.

An interval model of uncertainty may appear to be the most intuitive. However, there are significant advantages in accommodating also more expressive (and less pessimistic) uncertainty models. In [9], a financial portfolio optimization case-study is analyzed in which uncertainty arises from estimating the asset return rates. The authors claim that the interval model is too conservative in this scenario, because it would suggest to invest the whole capital into the asset with the smallest worst-case return. The ellipsoidal model proposed in that paper returns instead the more profitable strategy of spreading the capital across multiple assets. Further, depending on the field, researchers use different models to represent uncertainty. Maximum likelihood models are often used, for example, to estimate chemical reaction parameters [10]. To increase modeling expressiveness, we introduce the model of *Convex-MDP (CMDP)*, i.e., an MDP whose state transition probabilities are only known to lie within convex uncertainty sets. The proposed algorithms can be extended to verify CMDPs for all the models of uncertainty that satisfy a technical condition introduced later in the paper, while maintaining the same complexity results proven for IMDPs. This condition is not a limitation in practical scenarios, and we show that all the models in the wide and relevant class of convex uncertainty sets introduced in [11] (e.g. interval, ellipsoidal and likelihood models) satisfy it. Heterogeneous models of uncertainty can then be used within the same CMDP to represent different sources of uncertainty. We also note that the complexity results presented in [7] and [8] cannot be trivially extended to verifying CMDPs. This is because BFS are not defined for generic convex inequalities, so the construction of an equivalent MDP would not be possible. The complexity results are compared in Table 1.

To summarize, the contributions of this paper are as follows.

1. We give a polynomial-time algorithm for model checking PCTL properties (without *Bounded Until*) on IMDPs. This improves the co-NP result in [8] to PTIME.
2. We extend the algorithm to full PCTL (with *Bounded Until*) and show that its time complexity becomes pseudo-polynomial in the maximum integer bound in *Bounded Until*.
3. We show that our complexity results extend to Convex-MDPs (CMDPs) for a wide and expressive subclass of the convex models of uncertainty.
4. We demonstrate the relevance of our approach with case studies, where a small uncertainty in the probability transitions indeed yields a significant change in the verification results.

The paper is organized as follows. Section 2 gives background on MDPs, PCTL, and the analyzed uncertainty models. Section 3 presents related work in the fields of verification and control. Section 4 gives an overview of the proposed approach. In Section 5, we describe the proposed algorithm in detail and prove the PTIME complexity result. Section 6 describes two case studies, and we conclude and discuss future directions in Section 7.

2 Preliminaries

Definition 2.1. *A Probability Distribution (PD) over a finite set Z of cardinality n is a vector $\mu \in \mathbb{R}^n$ satisfying $\mathbf{0} \leq \mu \leq \mathbf{1}$ and $\mathbf{1}^T\mu = 1$. The element $\mu[i]$ represents the probability of realization of the event z_i. We call $Dist(Z)$ the set of distributions over Z.*

2.1 Convex Markov Decision Process (CMDP)

Definition 2.2. *A CMDP is a tuple $\mathcal{M}_\mathcal{C} = (S, S_0, A, \Omega, \mathcal{F}, \mathcal{A}, \mathcal{X}, L)$, where S is a finite set of states of cardinality $N = |S|$, S_0 is the set of initial states, A is a finite set of actions ($M = |A|$), Ω is a finite set of atomic propositions, $\mathcal{F}$ is a finite set of convex sets of transition PDs, $\mathcal{A} : S \to 2^A$ is a function that maps each state to the set of actions available at that state, $\mathcal{X} = S \times A \to \mathcal{F}$ is a function that associates to state s and action a the corresponding convex set $\mathcal{F}_s^a \in \mathcal{F}$ of transition PDs, and $L : S \to 2^\Omega$ is a labeling function.*

The set $\mathcal{F}_s^a = Dist_s^a(S)$ represents the uncertainty in defining a transition distribution for $\mathcal{M}_\mathcal{C}$ given state s and action a. We call $\mathbf{f}_s^a \in \mathcal{F}_s^a$ an observation of this uncertainty. Also, $\mathbf{f}_s^a \in \mathbb{R}^N$ and we can collect the vectors $\mathbf{f}_s^a, \forall s \in S$ into an observed transition matrix $F^a \in \mathbb{R}^{N\times N}$. Abusing terminology, we call $\mathcal{F}^a$ the uncertainty set of the transition matrices, and $F^a \in \mathcal{F}^a$. $\mathcal{F}_s^a$ is interpreted as the row of $\mathcal{F}^a$ corresponding to state s. Finally, $f_{s_i s_j}^a = \mathbf{f}_{s_i}^a[j]$ is the observed probability of transitioning from s_i to s_j when action a is selected.

A transition between state s to state s' in a CMDP occurs in three steps. First, an action $a \in \mathcal{A}(s)$ is chosen. The selection of a is nondeterministic. Secondly, an observed PD $\mathbf{f}_s^a \in \mathcal{F}_s^a$ is chosen. The selection of $\mathbf{f}_s^a$ models uncertainty in the transition. Lastly, a successor state s' is chosen randomly, according to the transition PD $\mathbf{f}_s^a$.

A path π in $\mathcal{M}_\mathcal{C}$ is a finite or infinite sequence of the form $s_0 \xrightarrow{f_{s_0 s_1}^{a_0}} s_1 \xrightarrow{f_{s_1 s_2}^{a_1}}, \cdots$, where $s_i \in S$, $a_i \in \mathcal{A}(s_i)$ and $f_{s_i,s_{i+1}}^{a_i} > 0 \; \forall i \geq 0$. We indicate with Π_{fin} (Π_{inf}) the set of all finite (infinite) paths of $\mathcal{M}_\mathcal{C}$. $\pi[i]$ is the i^{th} state along the path and, for finite paths, $last(\pi)$ is the last state visited in $\pi \in \Pi_{fin}$. $\Pi_s = \{\pi \mid \pi[0] = s\}$ is the set of paths starting in state s.

To model uncertainty in state transitions, we make the following assumptions:

Assumption 2.1. *$\mathcal{F}^a$ can be factored as the Cartesian product of its rows, i.e., its rows are uncorrelated. Formally, for every $a \in A$, $\mathcal{F}^a = \mathcal{F}_{s_0}^a \times \cdots \times \mathcal{F}_{s_{N-1}}^a$. In [11] this assumption is referred to as* rectangular uncertainty*.*

Assumption 2.2. *If the probability of a transition is zero (non-zero) for at least one PD in the uncertainty set, then it is zero (non-zero) for all PDs.*
Formally, $\exists \mathbf{f}_s^a \in \mathcal{F}_s^a : f_{ss'}^a = (\neq)0 \implies \forall \mathbf{f}_s^a \in \mathcal{F}_s^a : f_{ss'}^a = (\neq)0$. The assumption guarantees the correctness of the preprocessing verification routines used later in the paper, which rely on reachability of the states of the MDP underlying graph.

We determine the size $\mathcal{R}$ of the CMDP $\mathcal{M}_\mathcal{C}$ as follows. $\mathcal{M}_\mathcal{C}$ has N states, $O(M)$ actions per state and $O(N^2)$ transitions for each action. Let D_s^a denote the number of constraints required to express the rectangular uncertainty set $\mathcal{F}_s^a$ (e.g. $D_s^a = O(2N)$ for the interval model, to express the upper and lower bounds of the transition probabilities from state s to all states $s' \in S$), and $D = \max_{s\in S, a\in A} D_s^a$. The overall size of $\mathcal{M}_\mathcal{C}$ is thus $\mathcal{R} = O(N^2M + NMD)$.

In order to analyze *quantitative* properties of CMDPs, we need a probability space over infinite paths [12]. However, a probability space can only be constructed once nondeterminism and uncertainty have been resolved. We call each possible resolution of nondeterminism an *adversary*, which chooses an action in each state of $\mathcal{M}_\mathcal{C}$.

Definition 2.3. Adversary. *A randomized adversary for $\mathcal{M}_\mathcal{C}$ is a function $\alpha = \Pi_{fin} \times A \to [0,1]$, with $\sum_{\mathcal{A}(last(\pi))} \alpha(\pi, a) = 1$, and $a \in \mathcal{A}(last(\pi))$ if $\alpha(\pi, a) > 0$. We call Adv the set of all adversaries α of $\mathcal{M}_\mathcal{C}$.*

Conversely, we call a *nature* each possible resolution of uncertainty, i.e., a nature chooses a transition PD for each state and action of $\mathcal{M}_\mathcal{C}$.

Definition 2.4. Nature. *Given action $a \in A$, a randomized nature is the function $\eta^a : \Pi_{fin} \times Dist(S) \to [0,1]$ with $\int_{\mathcal{F}^a_{last(\pi)}} \eta^a(\pi, \mathbf{f}^a_s) = 1$, and $\mathbf{f}^a_s \in \mathcal{F}^a_{last(\pi)}$ if $\eta^a(\pi, \mathbf{f}^a_s) > 0$. We call Nat the set of all natures η^a of $\mathcal{M}_\mathcal{C}$.*

An adversary α (nature η^a) is memoryless if it depends only on $last(\pi)$. Also, α (η^a) is deterministic if $\alpha(\pi, a) = 1$ for some $a \in \mathcal{A}(last(\pi))$ ($\eta^a(\pi, f^a_s) = 1$ for some $\mathbf{f}^a_s \in \mathcal{F}^a_{last(\pi)}$).

2.2 Models of Uncertainty

We only consider CMDPs whose transition PDs lie in uncertainty sets that satisfy Assumption 5.1 (introduced later for ease of presentation). This assumption holds for all the uncertainty models analyzed in [11]. We report results for the interval, likelihood and ellipsoidal models. A more thorough derivation is available in Appendix A.

Interval Model. Intervals commonly describe uncertainty in transition matrices:

$$\mathcal{F}^a_s = \{\mathbf{f}^a_s \in \mathbb{R}^N \mid \mathbf{0} \leq \underline{\mathbf{f}}^a_s \leq \mathbf{f}^a_s \leq \overline{\mathbf{f}}^a_s \leq \mathbf{1}, \mathbf{1}^T \mathbf{f}^a_s = 1\} \tag{1}$$

where $\underline{\mathbf{f}}^a_s, \overline{\mathbf{f}}^a_s \in \mathbb{R}^N$ are the element-wise lower and upper bounds of $\mathbf{f}$. This model is suitable when the transition matrix components are individually estimated by statistical data.

Likelihood Model. This model is appropriate when the transition probabilities are determined experimentally. The transition frequencies associated to action $a \in A$ are collected in matrix H^a. Uncertainty in each row of H^a can be described by the likelihood region [13]:

$$\mathcal{F}^a_s = \{\mathbf{f}^a_s \in \mathbb{R}^N \mid \mathbf{f}^a_s \geq \mathbf{0}, \mathbf{1}^T \mathbf{f}^a_s = 1, \textstyle\sum_{s'} h^a_{ss'} \log(f^a_{ss'}) \geq \beta^a_s\} \tag{2}$$

where $\beta^a_s < \beta^a_{s,max} = \sum_{s'} h^a_{ss'} \log(h^a_{ss'})$ represents the uncertainty level. Likelihood regions are less conservative uncertainty representations than intervals, which arise from projections of the uncertainty region onto each row component.

Ellipsoidal Model. Ellipsoidal models can be seen as a second-order approximation of the likelihood model [11]. Formally:

$$\mathcal{F}^a_s = \{\mathbf{f}^a_s \in \mathbb{R}^N \mid \mathbf{f}^a_s \geq \mathbf{0}, \mathbf{1}^T \mathbf{f}^a_s = 1, \|R^a_s (\mathbf{f}^a_s - \mathbf{h}^a_s)\|_2 \leq 1, R^a_s \succ 0\} \tag{3}$$

where matrix R^a_s represents an ellipsoidal approximation of the likelihood Region (2).

Remark 2.1. Each set $\mathcal{F}^a_s$ within the same CMDP can be expressed with a different uncertainty model to represent different sources of uncertainty.

To illustrate our results, we will use the IMDP $\mathcal{M}_\mathcal{C}$ in Figure 1, with $S = \{s_0 \cdots s_3\}$, $S_0 = \{s_0\}$, $A = \{a, b\}$, $\Omega = \{\omega, \vartheta\}$, $\mathcal{A} : \{s_0, s_1, s_2\} \to \{a\}$; $\{s_3\} \to \{a, b\}$, $L : \{s_0, s_3\} \to \vartheta$; $\{s_2\} \to \omega$. The uncertainty intervals are shown next to each transition. For example, $\mathcal{F}^a_{s_0} = \{\mathbf{f}^a_{s_0} \in \mathbb{R}^N \mid [0, 0.6, 0.2, 0] \leq \mathbf{f}^a_{s_0} \leq [0, 0.8, 0.5, 0], \sum_{s' \in S} f^a_{ss'} = 1\}$.

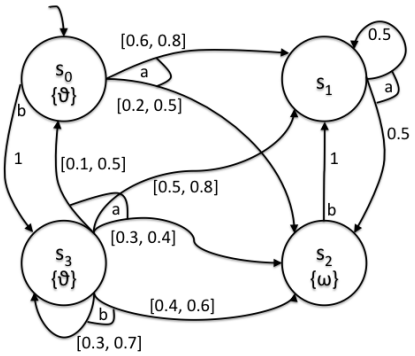

Fig. 1: Example IMDP.

Table 2: PCTL semantics for CMDP

$s \models \text{True}$	
$s \models \omega$	iff $\omega \in L(s)$
$s \models \neg\phi$	iff $s \not\models \phi$
$s \models \phi_1 \wedge \phi_2$	iff $s \models \phi_1 \wedge s \models \phi_2$
$s \models P_{\bowtie p}[\psi]$	iff $Prob(\{\pi \in \Pi_s(\alpha, \eta^a) \mid \pi \models \psi\}) \bowtie p$ $\forall \alpha \in Adv$ and $\eta^a \in Nat$
$\pi \models \mathcal{X}\phi$	iff $\pi[1] \models \phi$
$\pi \models \phi_1 \, \mathcal{U}^{\leq k} \phi_2$	iff $\exists i \leq k \mid \pi[i] \models \phi_2 \wedge \forall j < i \; \pi[j] \models \phi_1$
$\pi \models \phi_1 \, \mathcal{U} \phi_2$	iff $\exists k \geq 0 \mid \pi \models \phi_1 \, \mathcal{U}^{\leq k} \phi_2$

2.3 Probabilistic Computation Tree Logic (PCTL)

We use PCTL, a probabilistic logic derived from CTL which includes a probabilistic operator P [4], to express properties of CMDPs. The syntax of this logic is defined as follows:

$$\phi ::= True \mid \omega \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid P_{\bowtie p}[\psi] \qquad \text{state formulas}$$
$$\psi ::= \mathcal{X}\phi \mid \phi_1 \, \mathcal{U}^{\leq k} \phi_2 \mid \phi_1 \, \mathcal{U} \phi_2 \qquad \text{path formulas}$$

where $\omega \in \Omega$ is an atomic proposition, $\bowtie \in \{\leq, <, \geq, >\}$, $p \in [0, 1]$ and $k \in \mathbb{N}$.

Path formulas ψ use the *Next* ($\mathcal{X}$), *Bounded Until* ($\mathcal{U}^{\leq k}$) and *Unbounded Until* ($\mathcal{U}$) operators. These formulas are evaluated over paths and only allowed as parameters to the $P_{\bowtie p}[\psi]$ operator. The size $\mathcal{Q}$ of a PCTL formula is defined as the number of Boolean connectives plus the number of temporal operators in the formula. For the *Bounded Until* operator, we denote separately the maximum time bound that appears in the formula as k_{max}. Probabilistic statements about MDPs typically involve universal quantification over adversaries $\alpha \in Adv$. With uncertainties, for each action a selected by adversary α, we will further quantify across nature $\eta^a \in Nat$ to compute the worst case condition within the action range of η^a, i.e., the uncertainty set $\mathcal{F}_s^a$. We define $P_s(\alpha, \eta^a)[\psi] \triangleq Prob(\{\pi \in \Pi_s(\alpha, \eta^a) \mid \pi \models \psi\})$ the probability of taking a path $\pi \in \Pi_s$ that satisfies ψ under adversary α and nature η^a. If α and η^a are Markov deterministic in state s, we write $P_s(a, \mathbf{f}_s^a)$, where a and $\mathbf{f}_s^a$ are the action and resolution of uncertainty that are deterministically chosen at each execution step by α and η^a. $P_s^{max}[\psi]$ ($P_s^{min}[\psi]$) denote the maximum (minimum) probability $P_s(\alpha, \eta^a)[\psi]$ across all adversaries $\alpha \in Adv$ and natures $\eta^a \in Nat$, and the vectors $\mathbf{P}^{max}[\psi], \mathbf{P}^{min}[\psi] \in \mathbb{R}^N$ collect these probabilities $\forall s \in S$. The semantics of the logic is reported in Table 2, where we write $\models$ instead of $\models_{Adv,Nat}$ for simplicity.

For ease of computation, we would like to restrict our attention to memoryless and deterministic adversaries and natures to compute *quantitative* probabilities, i.e., solve problems:

$$P_s^{max}[\psi] = \max_{a \in \mathcal{A}(s)} \max_{\mathbf{f}_s^a \in \mathcal{F}_s^a} P_s(a, \mathbf{f}_s^a)[\psi] \qquad \text{or} \qquad P_s^{min}[\psi] = \min_{a \in \mathcal{A}(s)} \min_{\mathbf{f}_s^a \in \mathcal{F}_s^a} P_s(a, \mathbf{f}_s^a)[\psi] \tag{4}$$

We extend a result from [14] to prove that this is possible.

Proposition 2.1. *Given a CMDP $\mathcal{M}_\mathcal{C}$ and a target state $s_t \in S$, there always exist deterministic and memoryless adversaries and natures for $\mathcal{M}_\mathcal{C}$ that achieve the maximum (minimum) probabilities of reaching s_t, if A is finite and the inner optimization in Problem (4) always attains its optimum $\sigma_s^*(a)$ over the sets $\mathcal{F}_s^a, \forall s \in S, \forall a \in \mathcal{A}(s)$, i.e., there exists a finite feasible $\mathbf{f}_s^a \in \mathcal{F}_s^a$ such that $P_s(a, \mathbf{f}_s^a)[\psi] = \sigma_s^*(a)$.*

Sketch of proof. The proof is divided into two parts. First, we prove the existence of an adversary and a nature that achieve the maximum (minimum) probabilities of reaching t_s, using Banach fixed-point theorem [14]. Second, we prove that at least one such adversary and nature is memoryless and deterministic. The proof extends the one in Puterman [14], Theorem 6.2.10. We need to prove that Problem (4) always attains the maximum (minimum) over the feasibility sets $\mathcal{F}_s^a$, i.e., $\forall s \in S, \forall a \in \mathcal{A}(s), \exists \mathbf{f}_s^a \in \mathcal{F}_s^a : ||\mathbf{f}_s^a||_2 < \infty, P_s(a, \mathbf{f}_s^a)[\psi] = \sigma_s^*(a)$. This is indeed true for all the convex sets $\mathcal{F}_s^a$ considered in this paper. The interval and ellipsoidal models result in *compact* sets $\mathcal{F}_s^a$ on which Weierstrass theorem holds. For the likelihood model we use the notion of *consistency*, which guarantees the existence and uniqueness of a point in $\mathcal{F}_s^a$ where the optimum is attained. □

The verification of a PCTL state formula ϕ can be viewed as a decision problem. The verification algorithm V needs to determine whether a state $s \in S_0$ is (or is not) contained in the set $Sat(\phi) = \{s \in S \mid s \models \phi\}$. We can thus define the following properties for V:

Definition 2.5. Soundness *(*Completeness*). Algorithm V is sound (complete) if:*

$$s \in Sat_V(\phi) \Rightarrow s \in Sat(\phi) \qquad (s \notin Sat_V(\phi) \Rightarrow s \notin Sat(\phi))$$

where $Sat_V(\phi)$ is the computed satisfaction set, while $Sat(\phi)$ is the actual satisfaction set.

Algorithms to verify non-probabilistic formulas are sound and complete, because they are based on reachability analysis over the finite number of states of $\mathcal{M}_\mathcal{C}$ [15]. Conversely, we will show in Section 5 that algorithms to verify probabilistic formulas $\phi = P_{\bowtie p}[\psi]$ in the presence of uncertainties require to solve convex optimization problems over the set $\mathbb{R}$ of the real numbers. Optima of these problems can be arbitrary real numbers, so, in general, they can be computed only to within a desired accuracy ϵ_d. We consider an algorithm to be sound and complete if the error in determining the satisfaction probabilities of ϕ is bounded by such a parameter ϵ_d, since the returned result will still be accurate enough in most settings.

3 Related Work

Probabilistic model checking tools such as PRISM [5] have been used to analyze a multitude of applications, from communication protocols and biological pathways to security problems. In this paper, we further consider *uncertainties* in the probabilistic transitions of the MDP for model checking PCTL specifications. Prior work [6–8, 16] in similar verification problems also dealt with uncertainties in the probabilistic transitions. However, they considered only interval models of uncertainty, while we incorporate more expressive models such as ellipsoidal and likelihood. Further, we consider nature as adversarial and study how it affects the MDP execution in the worst case. The developers of PARAM [17] consider instead uncertainties as possible values that parameters in the model can take, and synthesize the optimal parameter values to maximize the satisfaction probability of a given PCTL specification.

We improve the previously best-known complexity result of co-NP in [8] to PTIME, for the fragment of PCTL without $\mathcal{U}^{\leq k}$. For the full PCTL syntax, our algorithm runs in $O(poly(\mathcal{R}) \times \mathcal{Q} \times k_{max})$ time, where k_{max} is the maximum bound in $\mathcal{U}^{\leq k}$. This result is pseudo-polynomial in k_{max}, i.e., polynomial (exponential) if k_{max} is counted in its unary (binary) representation. Conversely, classical PCTL model checking for DTMCs [4] runs in time polynomial in k_{max} counted in its binary representation. The difference stems from the computation of the set $Sat\left(P_{\bowtie p}\left[\phi_1 \, \mathcal{U}^{\leq k} \phi_2\right]\right)$. For (certain) MDPs, this computation involves raising the transition matrices $F^a, \forall a \in A$ to the k^{th} power, to model the evolution of the system in k steps. With uncertainties, we cannot do matrix exponentiation, because

$F^a \in \mathcal{F}^a$ might change at each step. However, both $\mathcal{Q}$ and k_{max} are typically small in practical applications [18,19], so the dominant factor for runtime is the size of the model $\mathcal{R}$. We note that the complexity results of [7] and [8] can be extended to the PCTL with $\mathcal{U}^{\leq k}$.

The convex uncertainty models [11] analyzed in this paper have been considered recently in the robust control literature. In [20], an algorithm is given to synthesize a robust optimal controller for an MDP to satisfy a Linear Temporal Logic (LTL) specification where only one probabilistic operator is allowed. Their technique first converts the LTL specification to a Rabin automaton (which is worst-case doubly exponential in the size of the LTL formula), and composes it with the MDP. Robust dynamic programming is then used to solve for the optimal control policy. We consider PCTL, which allows nested probability operators, and propose an algorithm which is polynomial both in the size of the model and of the formula.

In [21], the robustness of PCTL model checking is analyzed based on the notion of an Approximate Probabilistic Bisimulation (APB) tailored to the finite-precision approximation of a numerical model. We instead verify MDPs whose transition probabilities are affected by uncertainties due to estimation errors or imperfect information about the environment.

4 Probabilistic Model Checking with Uncertainties

We define the problem under analysis, and overview the proposed approach to solve it.

PCTL model checking with uncertainties. **Given** a Markov Decision Process model with convex uncertainties $\mathcal{M}_{\mathcal{C}}$ of size $\mathcal{R}$ and a PCTL formula ϕ of size $\mathcal{Q}$ over a set of atomic propositions Ω, **verify** ϕ over the uncertainty sets $\mathcal{F}_s^a \in \mathcal{F}$ of $\mathcal{M}_{\mathcal{C}}$.

As in verification of CTL [22], the algorithm traverses bottom-up the parse tree for ϕ, recursively computing the set $Sat(\phi')$ of states satisfying each sub-formula ϕ'. At the end of the traversal, the algorithm computes the set of states satisfying ϕ and it determines if $s \models \phi$ by checking if $s \in Sat\,(\phi)$. For the non-probabilistic PCTL operators, the satisfying states are computed as: $Sat\,(True) = S$, $Sat(\omega) = \{s \in S \mid \omega \in L(s)\}$, $Sat(\neg\phi) = S \setminus Sat(\phi)$ and $Sat(\phi_1 \wedge \phi_2) = Sat(\phi_1) \cap Sat(\phi_2)$. For the probabilistic operator $P \bowtie [\psi]$, we compute:

$$Sat\,(P_{\lhd p}\,[\psi]) = \{s \in S \mid P_s^{max}(\psi) \lhd p\} \qquad Sat\,(P_{\rhd p}\,[\psi]) = \left\{s \in S \mid P_s^{min}(\psi) \rhd p\right\} \tag{5}$$

In this paper, we propose polynomial-time routines to compute Sets 5 for MDPs whose transition matrices F^a are only known to lie within convex uncertainty sets $\mathcal{F}^a$, $\forall a \in A$.

Using Proposition 2.1, the proposed routines encode the transitions of $\mathcal{M}_{\mathcal{C}}$ under the sets of deterministic and memoryless adversaries and natures into convex programs and solve them. From the returned solution, it is then possible to determine the *quantitative* satisfaction probabilities $P_s^{max}[\psi]$ (or $P_s^{min}[\psi]$) $\forall s \in S$, which get compared in linear time to the threshold p to compute the set $Sat\,(P_{\bowtie p}\,[\psi])$. To prove the polynomial-time complexity of the model-checking algorithm, we use the following key result from convex theory [23].

Proposition 4.1. *Given the convex program:*

$$\begin{aligned} &\min_{\mathbf{x}} f_0(\mathbf{x}) \\ &s.t.\ f_i(\mathbf{x}) \leq 0 \qquad\qquad i = 1, \cdots, m \end{aligned}$$

with $\mathbf{x} \in \mathbb{R}^n$ *and* $f_i, i = 0, \cdots, m$ *convex functions, the optimum* σ^* *can be found to within* $\pm\epsilon_d$ *in time complexity that is polynomial in the size of the problem* (n, m) *and* $\log(1/\epsilon_d)$.

We are now ready to state the main contribution of this paper:

Theorem 4.1. Complexity of PCTL model-checking for CMDPs.

1. *The problem of verifying if a CMDP $\mathcal{M}_\mathcal{C}$ of size $\mathcal{R}$ satisfies a PCTL formula ϕ without $\mathcal{U}^{\leq k}$ is in PTIME.*
2. *A formula ϕ' with $\mathcal{U}^{\leq k}$ can be verified with time complexity $O\left(poly(\mathcal{R}) \times \mathcal{Q}' \times k_{max}\right)$, i.e., pseudo-polynomial in the maximum time bound k_{max} of $\mathcal{U}^{\leq k}$.*

Sketch of proof. The proof is constructive. Our verification algorithm parses ϕ in time linear in the size $\mathcal{Q}$ of ϕ [22], computing the satisfiability set of each operator in ϕ. For the non-probabilistic operators, satisfiability sets can be computed in time polynomial in $\mathcal{R}$ using set operations, i.e., set inclusion, complementation and intersection. For the probabilistic operator, we leverage Proposition 4.1 and prove that the proposed verification routines: 1) solve a number of convex problems polynomial in $\mathcal{R}$; 2) generate these convex programs in time polynomial in $\mathcal{R}$. The correctness and time-complexity for formulas involving *Next* and *Unbounded Until* operators are formalized in Lemma 5.1 and 5.2 (Section 5.1 and 5.2). It thus follows that the overall algorithm runs in time polynomial in $\mathcal{R}$ and in the size of ϕ. Finally, Lemma 5.3 formalizes the results related to the *Bounded Until* operator. □

5 Verification Routines

We detail the routines used to verify the probabilistic operator P. We only consider properties in the form $\phi = P_{\lhd p}[\psi]$, but the results can trivially be extended to $\phi = P_{\rhd p}[\psi]$ by replacing "max" with "min" in the optimization problems below.

5.1 *Next* Operator

We verify property $\phi = P_{\bowtie p}[\mathcal{X}\phi_1]$ on a CMDP of size $\mathcal{R}$. First, the set $S^{yes} = Sat(\phi_1)$ is computed. Second, for all state $s \in S$, the algorithm evaluates Equation (4) by solving:

$$P_s^{max}[\mathcal{X}\phi_1] = \max_{a \in \mathcal{A}(s)} \max_{\mathbf{f}_s^a \in \mathcal{F}_s^a} \sum_{s' \in S^{yes}} f_{ss'}^a \tag{6}$$

The inner max is a convex program since $\mathcal{F}_s^a$ is convex. The sets $\mathcal{F}_s^a$ can be expressed with heterogeneous uncertainty models, since each problem is independent from the others. Finally, the computed probabilities are compared to p to select the states that satisfy ϕ.

Lemma 5.1. *The routine to verify the* Next *operator is sound, complete and guaranteed to terminate with algorithmic complexity that is polynomial in the size $\mathcal{R}$ of $\mathcal{M}_\mathcal{C}$.*

Proof. Problem (6) has one "inner" convex program $\forall s \in S$ and $\forall a \in \mathcal{A}(s)$, for a total of $O(MN)$ problems. Each problem has $O(N)$ unknowns, representing the probability of transitioning from state s to state s' for $s' \in S^{yes}$. It has $O(N+1)$ constraints to guarantee that the solution lies in the probability simplex, and D_s^a constraints to enforce the solution to be within the uncertainty set $\mathcal{F}_s^a$. The total number of constraints is thus linear in $\mathcal{R}$. Using Proposition 4.1, each inner problem is solved in time polynomial in $\mathcal{R}$. Soundness and completeness also follow directly from Proposition 4.1, which states that the optimum of Problem (6) can be found to within the desired accuracy $\pm\epsilon_d$ in time linear in $\log(1/\epsilon_d)$. □

We verify $\phi = P_{\leq 0.4}[\mathcal{X}\omega]$ in the example in Figure 1. Trivially, $S^{yes} = \{s_2\}$. We solve Problem (6) for all $a \in A$ and $s \in S$. As an example, for state s_0 and action a, we solve:

$$P_{s_0}^{a,max} = \max_{f_{01}, f_{02}} f_{02}$$
$$\text{s.t. } 0.6 \leq f_{01} \leq 0.8;\ 0.2 \leq f_{02} \leq 0.5;\ f_{01} + f_{02} = 1$$

and get $P_{s_0}^{a,max}[\mathcal{X}\omega] = 0.4$. Overall, we get $\mathbf{P}^{max}[\mathcal{X}\omega]$=$[0.4, 0.5, 0, 0.6]$, so $Sat(\phi)$=$\{s_0, s_2\}$.

5.2 *Unbounded Until* Operator

We verify $\phi = P_{\lhd p}[\phi_1 \mathcal{U} \phi_2]$ on a CMDP of size $\mathcal{R}$. First, the sets $S^{yes} \triangleq Sat\left(P_{\geq 1}[\phi_1 \mathcal{U} \phi_2]\right)$, $S^{no} \triangleq Sat\left(P_{\leq 0}[\phi_1 \mathcal{U} \phi_2]\right)$ and $S^{?} = S \setminus (S^{no} \cup S^{yes})$ are precomputed in time polynomial in $\mathcal{R}$ using conventional reachability routines over the CMDP underlying graph [15]. Second, Equation (4) is evaluated for all $s \in S$ at the same time using the Convex Programming procedure described next. Finally, the computed probabilities are compared to p. An alternative verification procedure, based on Value Iteration, can be found in Appendix B.

Convex Programming Procedure (CP). We start from the classical LP formulation to solve the problem without the presence of uncertainty [15]:

$$\begin{aligned} &\min_{\mathbf{x}} \mathbf{x^T 1} \\ &\text{s.t. } x_s = 0;\ x_s = 1; && \forall s \in S^{no};\ s \in S^{yes}; \\ &\quad x_s \geq \mathbf{x^T f_s^a} && \forall s \in S^{?}, \forall a \in \mathcal{A}(s) \end{aligned} \tag{7}$$

where $\mathbf{P}^{max}[\phi_1 \mathcal{U} \phi_2] = \mathbf{x}^*$ is computed solving only one LP. Problem (7) has N unknowns and $N - Q + MQ$ constraints, where $Q = |S^{?}| = O(N)$, so its size is polynomial in $\mathcal{R}$.

Proposition 2.1 allows us to rewrite Problem (7) in the uncertain scenario as:

$$\begin{aligned} &\min_{\mathbf{x}} \mathbf{x^T 1} \\ &\text{s.t. } x_s = 0;\ x_s = 1; && \forall s \in S^{no};\ \forall s \in S^{yes}; \\ &\quad x_s \geq \max_{\mathbf{f_s^a} \in \mathcal{F}_s^a} \left(\mathbf{x^T f_s^a}\right) && \forall s \in S^{?}, \forall a \in \mathcal{A}(s) \end{aligned} \tag{8}$$

i.e., we maximize the lower bound on x_s across the nature action range. The decision variable of the inner problem is $\mathbf{f}_s^a$ and its optimal value $\sigma^*(\mathbf{x})$ is parameterized in the outer problem decision variable $\mathbf{x}$. Problem (8) can be written in convex form for an arbitrary uncertainty model by replacing the last constraint with one constraint for each point in $\mathcal{F}_s^a$. However, this approach results in infinite constraints if the set $\mathcal{F}_s^a$ contains infinitely many points, as in the cases considered in the paper. We solve this difficulty using duality, which allows to rewrite Problem (8) with a number of constraints polynomial in $\mathcal{R}$. We start by replacing the primal inner problem in the outer Problem (8) with its dual $\forall s \in S^{?}$ and $\forall a \in \mathcal{A}(s)$:

$$\sigma_s^a(\mathbf{x}) = \max_{\mathbf{f}_s^a \in \mathcal{F}_s^a} \mathbf{x}^T \mathbf{f}_s^a \qquad \Rightarrow \qquad d_s^a(\mathbf{x}) = \min_{\lambda_{\mathbf{s}}^{\mathbf{a}} \in \mathcal{D}_s^a} g(\lambda_{\mathbf{s}}^{\mathbf{a}}, \mathbf{x}) \tag{9}$$

where $\lambda_{\mathbf{s}}^{\mathbf{a}}$ is the (vector) Lagrange multiplier and $\mathcal{D}_s^a$ is the feasibility set of the dual. In the dual, the decision variable is $\lambda_{\mathbf{s}}^{\mathbf{a}}$ and its optimal value $d_s^a(\mathbf{x})$ is parameterized in $\mathbf{x}$. The dual function $g(\lambda_s^a, \mathbf{x})$ and the set $\mathcal{D}_s^a$ are convex by construction in λ_s^a for arbitrary uncertainty models, so the dual is convex. Further, since also the primal is convex, strong duality holds, i.e., $\sigma_s^a = d_s^a$, $\forall \mathbf{x} \in \mathbb{R}^N$, because the primal satisfies Slater's condition [24] for any non-trivial uncertainty set $\mathcal{F}_s^a$. Any dual solution overestimates the primal solution. When substituting the primals with the duals in Problem (8), we drop the inner optimization operators because the outer optimization operator will find the least overestimates, i.e., the dual solutions $d_s^a, \forall s \in S, a \in \mathcal{A}(s)$, to minimize its cost function. We get the CP formulation:

$$\begin{aligned} &\min_{\mathbf{x}} \mathbf{x^T 1} && && \min_{\mathbf{x},\lambda} \mathbf{x^T 1} \\ &\text{s.t. } x_s = 0;\ x_s = 1; && && \text{s.t. } x_s = 0;\ x_s = 1; && \forall s \in S^{no}; \forall s \in S^{yes}; && \text{(10a)} \\ &\quad x_s \geq \min_{\lambda_s^a \in \mathcal{D}_s^a} g\left(\lambda_s^a, \mathbf{x}\right) && \Rightarrow && \quad x_s \geq g\left(\lambda_s^a, \mathbf{x}\right); && \forall s \in S^{?}, \forall a \in \mathcal{A}(s); && \text{(10b)} \\ & && && \quad \lambda_s^a \in \mathcal{D}_s^a && \forall s \in S^{?}, \forall a \in \mathcal{A}(s) && \text{(10c)} \end{aligned}$$

The decision variables of Problem (10) are both $\mathbf{x}$ and λ_s^a, so the CP formulation is convex only if the dual function $g(\lambda_s^a, \mathbf{x})$ is jointly convex in λ_s^a and $\mathbf{x}$. While this condition cannot be guaranteed for arbitrary uncertainty models, we prove constructively that it holds for the ones considered in the paper. For example, for the interval model, Problem (10) reads:

$$\begin{aligned}
&\min_{\mathbf{x}, \lambda_{\mathbf{s}}^{\mathbf{a}}} \mathbf{x^T 1} \\
&\text{s.t. } x_s = 0;\ x_s = 1; && \forall s \in S^{no}; \forall s \in S^{yes}; \\
&\quad x_s \geq \lambda_{1,s}^a - (\underline{\mathbf{f}}_a^s)^T \lambda_{\mathbf{2,s}}^{\mathbf{a}} + (\overline{\mathbf{f}}_a^s)^T \lambda_{\mathbf{3,s}}^{\mathbf{a}}; && \forall s \in S^?, \forall a \in \mathcal{A}(s); \\
&\quad \mathbf{x} + \lambda_{\mathbf{2,s}}^{\mathbf{a}} - \lambda_{\mathbf{3,s}}^{\mathbf{a}} - \lambda_{1,s}^a \mathbf{1} = \mathbf{0}; && \forall s \in S^?, \forall a \in \mathcal{A}(s); \\
&\quad \lambda_{\mathbf{2,s}}^{\mathbf{a}} \geq \mathbf{0},\ \lambda_{\mathbf{3,s}}^{\mathbf{a}} \geq \mathbf{0} && \forall s \in S^?, \forall a \in \mathcal{A}(s)
\end{aligned}$$

which is an LP, so trivially jointly convex in $\mathbf{x}$ and λ_s^a. Analogously, Problem (10) for the ellipsoidal model is a Second-Order Cone Program (SOCP), as reported in Appendix A, so again jointly convex in $\mathbf{x}$ and λ_s^a. For the likelihood model, Problem (10) reads:

$$\begin{aligned}
&\min_{x_s, \lambda_s^a} \mathbf{x_s^T 1} \\
&\text{s.t. } x_s = 0;\ x_s = 1; && \forall s \in S^{no}; \forall s \in S^{yes}; && \text{(11a)} \\
&\quad x_s \geq \lambda_{1,s}^a - (1+\beta_s^a)\lambda_{2,s}^a + \lambda_{2,s}^a \textstyle\sum_{s'} h_{ss'}^a \log\left(\frac{\lambda_{2,s}^a h_{ss'}^a}{\lambda_{1,s}^a - x_{s'}}\right); && \forall s \in S^?, \forall a \in \mathcal{A}(s); && \text{(11b)} \\
&\quad \lambda_{1,s}^a \geq \max_{s' \in S} x_{s'};\ \lambda_{2,s}^a \geq 0 && \forall s \in S^?, \forall a \in \mathcal{A}(s) && \text{(11c)}
\end{aligned}$$

We prove its joint convexity in $\mathbf{x}$ and λ_s^a as follows. The cost function and Constraints (11a)-(11c) are trivially convex. Constraint (11b) is generated by a primal-dual transformation, so, according to convex theory, it is convex in the dual variables λ_s^a by construction. Convex theory also guarantees that the affine subtraction of $\mathbf{x}$ from $\lambda_{1,s}^a$ preserves convexity, given $\lambda_{1,s}^a \geq \max x_{s'}, \forall s \in S$ in Constraint (11c), so we conclude that Problem (11) is convex.

For general CMDPs, we will assume:

Assumption 5.1. *Given a CMDP $\mathcal{M}_\mathcal{C}$, for all convex uncertainty sets $\mathcal{F}_s^a \in \mathcal{F}$, the dual function $g(\lambda_s^a, \mathbf{x})$ in Problem (9) is jointly convex in both $\lambda_{\mathbf{s}}^{\mathbf{a}}$ and $\mathbf{x}$.*

According to Proposition 4.1, Problem (10) can thus be solved in polynomial time. Also for this formulation, $\mathbf{P}^{max}[\phi_1 \mathcal{U} \phi_2] = \mathbf{x}^*$, so all the satisfaction probabilities can be computed by solving only one convex problem. Finally, we note that we can combine models of uncertainty different from one another within a single CP formulation, since each dual problem is independent from the others according to Assumption 2.1. As an example, if both the interval and ellipsoidal models are used, the overall CP formulation is an SOCP.

Lemma 5.2. *The routine to verify the* Unbounded Until *operator is sound, complete and guaranteed to terminate with algorithmic complexity polynomial in the size $\mathcal{R}$ of $\mathcal{M}_\mathcal{C}$, if $\mathcal{M}_\mathcal{C}$ satisfies Assumption 5.1.*

Proof. The routine solves only one convex program, generated in time polynomial in $\mathcal{R}$ as follows. We formulate Constraints (10b) and (10c) $\forall s \in S^?$ and $a \in \mathcal{A}(s)$, i.e., $O(MQ)$ constraints, where $Q = |S^?| = O(N)$. They are derived from MQ primal-dual transformations as in Equation (9). Each primal problem has N unknowns, $N+1$ constraints to represent the probability simplex and D_s^a constraints to represent the uncertainty set $\mathcal{F}_s^a$. From duality theory, the corresponding dual inner problem has $N+1+D_s^a$ unknowns and $2N+1+D_s^a$ constraints. Overall, Problem (10) has $O\left((N+1+D)MQ\right)$ more unknowns and $O\left((2N+1+D)MQ\right)$ more constraints of Problem (7), so its size is polynomial in $\mathcal{R}$.

If $\mathcal{M}_\mathcal{C}$ satisfies Assumption 5.1, Problem (10) is convex. Using Proposition 4.1, we conclude that it can be solved in time polynomial in $\mathcal{R}$. Finally, when strong duality holds for the transformation in Equation (9), soundness and completeness of the final solution are preserved because the dual and primal optimal value of each inner problem are equivalent. □

We verify $\phi = P_{\geq 0.3}[\, \vartheta \, \mathcal{U} \, \omega \,]$ on the example in Figure 1. Problem (10) written with the data of the model has 19 variables and 11 constraints, and it can be found in Appendix C. The solution reads: $\mathbf{P}^{min}[\, \vartheta \, \mathcal{U} \, \omega \,] = [0.2, 0, 1, 0.32]$, and, in conclusion, $Sat(\phi) = \{s_2, s_3\}$.

5.3 *Bounded Until* Operator

We present the routine to verify property $\phi = P_{\bowtie p}[\phi_1 \mathcal{U}^{\leq k} \phi_2]$ on a CMDP of size $\mathcal{R}$. First, the set $S^{yes} \stackrel{def}{=} Sat(\phi_2)$, $S^{no} \stackrel{def}{=} S \setminus (Sat(\phi_1) \cup Sat(\phi_2))$ and $S^{?} = S \setminus (S^{no} \cup S^{yes})$ are precomputed. Second, the maximum probabilities $\mathbf{P}^{max}[\psi] = \mathbf{x}^k = G^k(\mathbf{0})$ to satisfy ϕ are computed for all states $s \in S$ applying k times mapping G defined as:

$$\mathbf{x}^i = G^i(\mathbf{x}^{i-1}) = \begin{cases} 0;\ 1; & \forall s \in S^{no};\ \forall s \in S^{yes}; \\ 0; & \forall s \in S^{?} \wedge i = 0; \\ \max\limits_{a \in \mathcal{A}(s)} \max\limits_{\mathbf{f}_s^a \in \mathcal{F}_s^a} (\mathbf{x}^{i-1})^T \mathbf{f}_\mathbf{s}^\mathbf{a} & \forall s \in S^{?} \wedge i > 0 \end{cases} \tag{12}$$

and $\mathbf{x}^{-1} = \mathbf{0} \in \mathbb{R}^N$. Finally, the computed probabilities are compared to the threshold p.

Lemma 5.3. *The routine to verify the* Bounded Until *operator is sound, complete and guaranteed to terminate with algorithmic complexity that is polynomial in the size $\mathcal{R}$ of $\mathcal{M}_\mathcal{C}$ and pseudo-polynomial in the time bound k of $\mathcal{U}^{\leq k}$.*

Proof. The proof of polynomial complexity in $\mathcal{R}$ is similar to the one for the *Next* Operator. Further, the pseudo-polynomial complexity in k comes from applying Mapping (12) k times. While each inner problem is solved with accuracy $\pm\epsilon_{inn}$ in time linear in $\log(1/\epsilon_{inn})$ by Proposition 4.1, we also need to prove the soundness and completeness of the overall solution, since the ϵ_{inn}-approximations in $\mathbf{x}^i, \forall i$, get propagated at each iteration and the error might get amplified. We call ϵ_s^i the error accumulated at step i for state s, $x_s^i = x_{s,id}^i + \epsilon_s^i$, where $x_{s,id}^i$ is the solution with no error propagation, and ϵ_s^k the error in the final solution. Also, $\mathbf{f}_s^{a,i} \in \mathcal{F}_s^a$ is the optimal solution of the inner problem at step i. We solve this difficulty by noting that the optimal value of the inner problem is computed by multiplying vector $\mathbf{x}^i$ by $\mathbf{f}_s^{a,i} \in \mathcal{F}_s^a$, with $\mathbf{1}^T \mathbf{f}_s^a = 1, \forall \mathbf{f}_s^a \in \mathcal{F}_s^a, \forall a \in \mathcal{A}(s)$. At the first, second and i^{th} iteration:

$$\begin{aligned} x_s^1 &= x_{s,id}^1 + \epsilon_s^1 = \mathbf{f}_s^{a,1}\mathbf{x}^0 + \epsilon_{inn} \\ x_s^2 &= \mathbf{f}_s^{a,2}\mathbf{x}^1 + \epsilon_{inn} = \mathbf{f}_s^{a,2}\left(\mathbf{f}_s^{a,1}\mathbf{x}^0 + \epsilon_{inn}\mathbf{1}\right) + \epsilon_{inn} = \mathbf{f}_s^{a,2}\mathbf{f}_s^{a,1}\mathbf{x}^0 + 2\epsilon_{inn} \\ x_s^i &= \mathbf{f}_s^{a,i}\mathbf{x}^{i-1} + \epsilon_{inn} = \mathbf{f}_s^{a,i}(\mathbf{f}_s^{a,i-1}\mathbf{x}^{i-1} + (i-1)\epsilon_{inn}\mathbf{1}) + \epsilon_{inn} = \mathbf{f}_s^{a,i}\mathbf{f}_s^{a,i-1}\ldots\mathbf{f}_s^{a,1}\mathbf{x}^0 + i\epsilon_{inn} \end{aligned}$$

so ϵ_s^i increases linearly with i. The desired accuracy ϵ_d of the final solution can thus be guaranteed by solving each inner problem with accuracy $\epsilon_{inn} = \epsilon_d/k$. □

We verify $\phi = P_{\leq 0.6}[\vartheta \, \mathcal{U}^{\leq 1} \omega]$ in the example in Fig. 1. $S^{yes} = \{s_2\}$, $S^{no} = \{s_1\}$. Applying once Mapping (12), we get $\mathbf{P}^{max}[\vartheta \mathcal{U}^{\leq 1} \omega] = [0.4, 0, 1, 0.6]$ and $Sat(\phi) = \{s_0, s_1, s_3\}$.

6 Case Studies

We implemented the proposed verification algorithm in Python, and interfaced it with PRISM [5] to extract information about the CMDP model. We used MOSEK [25] to solve

the LPs generated for the interval model and implemented customized numerical solvers for the other models of uncertainty. The implemented tool is available at [26]. The algorithm was tested on all the case studies collected in the PRISM benchmark suite [27]. Due to space limits, we report two of them: the verification of a consensus protocol and of a dynamic configuration protocol for IPv4 addresses. Further, the Dining Philosopher problem is verified in Appendix D. The goals of these experiments are two-fold: 1) quantitatively evaluate the impact of uncertainty on the results of verification of PCTL properties of CMDPs; 2) assess the scalability of the proposed approach to increasing problem size. The runtime data were obtained on a 2.4 GHz Intel Xeon with 32GB of RAM.

6.1 Consensus Protocol

Consensus problems arise in many distributed environments, where a group of distributed processes attempt to reach an agreement about a decision to take by accessing some shared entity. A consensus protocol ensures that the processes will eventually terminate and take the same decision, even if they start with initial guesses that might differ from one another.

We analyze the randomized consensus protocol presented in [18, 28]. The protocol guarantees that the processes return a preference value $v \in \{1, 2\}$, with probability parameterized by a process independent value R ($R \geq 2$) and the number of processes P. The processes communicate with one another by accessing a shared counter of value c. The protocol proceeds in rounds. At each round, a process flips a local coin, increments or decrements the shared counter depending on the outcome and then reads its value c. If $c \geq PR$ ($c \leq -PR$), it chooses $v = 1$ ($v = 2$). Note that the larger the value of R, the longer it takes on average for the processes to reach the decision. Nondeterminism is used to model the asynchronous access of the processes to the shared counter, so the overall protocol is modeled as an MDP.

We verify the property **Agreement:** all processes must agree on the same decision, i.e., choose a value $v \in \{1, 2\}$. We compute the minimum probability of **Agreement** and compare it against the theoretical lower bound $(R-1)/2R$ [18]. In PCTL syntax:

$$P_{s_0}^{min}[\psi] := P_{s_0}^{min}(\mathbf{F}(\{finished\} \wedge \{all\_coins\_equal\_1\})) \tag{13}$$

We consider the case where one of the processes is unreliable or adversarial, i.e., it throws a biased coin instead of a fair coin. Specifically, the probability of either outcome lies in the uncertainty interval $[(1-u)p_0, (1+u)p_0]$, where $p_0 = 0.5$ according to the protocol. This setting is relevant to analyze the protocol robustness when a process acts erroneously due to a failure or a security breach. In particular, our approach allows to study attacks that deliberately hide under the noise threshold of the protocol. In such attacks, the compromised node defers agreement by producing outputs whose statistical properties are within the noise tolerance of an uncompromised node, so that it is harder to detect its malicious behavior.

Figure 2 shows the effect of different levels of uncertainty on the computed probabilities for $P = 4$. With no uncertainty ($u = 0$), $P_{s_0}^{min}$ increases as R increases, because a larger R drives the decision regions further apart, making it more difficult for the processes to decide on different values of v. As R goes to infinity, $P_{s_0}^{min}$ approaches the theoretical lower bound $lim_{R\to\infty}(R-1)/2R = 0.5$. However, even with a small uncertainty ($u = 0.01$), $P_{s_0}^{min}$ soon decreases for increasing R. With a large uncertainty ($u = 0.15$), $P_{s_0}^{min}$ quickly goes to 0. A possible explanation is that the faulty process has more opportunities to deter agreement for a high R, since R also determines the expected time to termination. Results thus show that the protocol is vulnerable to uncertainties. This fact may have serious security implication, i.e., a denial-of-service attack could reduce the availability of the distributed service, since a compromised process may substantially alter the expected probability of agreement.

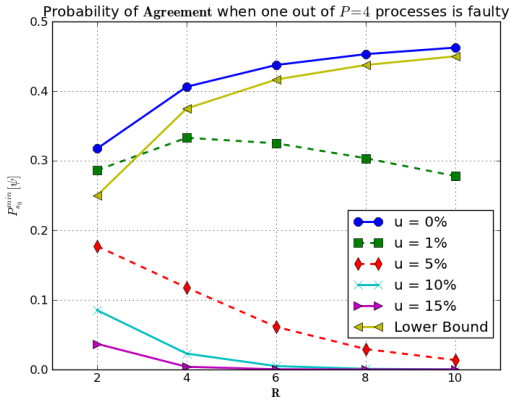

Fig. 2: Value of Eq. 13 in function of R while varying the uncertainty level u.

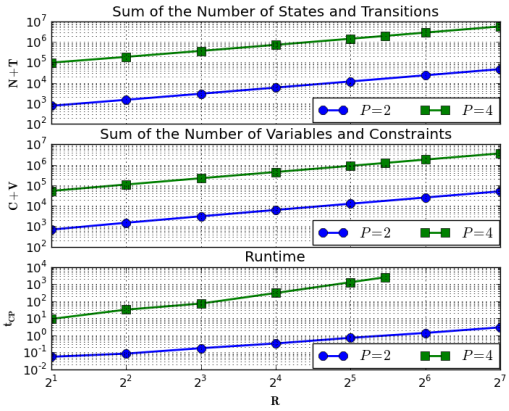

Fig. 3: Scalability of the CP procedure.

Lastly, we study the scalability of the CP procedure, by evaluating Equation (13) while sweeping R both for $P = 2$ and $P = 4$. We use MOSEK [25] to solve Problem (10) and set the Time Out (TO) to one hour. In Figure 3, we plot the sum $(N + T)$ of the number of states (N) and transitions (T) of the CMDP, which are independent of the uncertainty in the transition probabilities, to represent the model size (top), the sum $(V+C)$ of the number of variables (V) and constraints (C) of the generated LP instances of Problem (10) (center), and the running time t_{CP} (bottom). $V + C$ always scales linearly with $N + T$ (the lines have the same slope), supporting the polynomial complexity result for our algorithm. Instead, t_{CP} scales linearly only for smaller problems $(P = 2)$, while it has a higher-order polynomial behavior for larger problems $(P = 4)$ (the line is still a straight line but with steeper slope, so it is polynomial on logarithmic axes). This behavior depends on the performance of the chosen numerical solver, and it can improve benefiting of future advancements in the solver implementation. In Table 3, we compare the CP procedure with two tools, PRISM [5] and PARAM [17], in terms of runtime, for varying values of P and R. Although neither tool solves the same problem addressed in this paper, the comparison is useful to assess the practicality of the proposed approach. In particular, PRISM only verifies PCTL properties of MDPs with no uncertainties. PARAM instead derives a symbolic expression of the satisfaction probabilities as a function of the model parameters, to then find the parameter values that satisfy the property. Hence, PRISM only considers a special case of the models considered in this paper, while our approach only returns the worst-case scenario computed by PARAM. Results show that the CP procedure runs faster than PRISM for some benchmarks, but it is slower for larger models. This is expected since the scalability of our approach depends mainly on the problem size, while the performance of the iterative engine in PRISM depends on the problem size and on the number of iterations required to achieve convergence, which is dependent on the problem data. Finally, our approach is orders of magnitude faster than PARAM, so it should be preferred to perform worst-case analysis of system performances.

6.2 ZeroConf Dynamic Configuration Protocol for IPv4 Link-Local Addresses

The ZeroConf protocol [29, 30] is an Internet Protocol (IP)-based configuration protocol for local (e.g. domestic) networks. In such a local context, each device should configure its own unique IP address when it gets connected to the network, with no user intervention. The protocol thus offers a distributed "plug-and-play" solution in which address configuration is managed by individual devices when they are connected to the network. The network is

Table 3: Runtime Comparison

Tool	$P=2,R=2$	$R=7$	$R=128$	$P=4,R=2$	$R=32$	$R=44$	$P=6,R=4$
	$N+T=764$	$2,604$	$47,132$	$97,888$	$1,262,688$	$1,979,488$	$14,211,904$
CP	$0.02s$	$0.1s$	$2.1s$	$8.3s$	$1,341s$	$2,689$	TO
PRISM	$0.01s$	$0.09s$	$196s$	$1s$	$2,047s$	TO	$1860s$
PARAM	$22.8s$	$657s$	TO	TO	TO	TO	TO

composed of DV_{tot} devices. After being connected, a new device chooses randomly an IP address from a pool of $IP_A = 65024$ available ones, as specified by the standard. The address is non-utilized with probability $p_0 = 1 - DV_{tot}/IP_A$. It then sends messages to the other devices in the network, asking whether the chosen IP address is already in use. If no reply is received, the device starts using the IP address, otherwise the process is repeated.

The protocol is both probabilistic and timed: probability is used in the randomized selection of an IP address and to model the eventuality of message loss; timing defines intervals that elapse between message retransmissions. In [30], the protocol has been modeled as an MDP using the digital clock semantic of time. In this semantic, time is discretized in a finite set of epochs which are mapped to a finite number of states in an MDP, indexed by the epoch variable t_e. To enhance the user experience and, in battery-powered devices, to save energy, it is important to guarantee that a newly-connected device manages to select a unique IP address within a given deadline *dl*. For numerical reasons, we study the maximum probability of *not* being able to select a valid address within *dl*. In PCTL syntax:

$$P_{s_0}^{max}\left[\psi\right] := P_{s_0}^{max}\left(\neg\{unique\_address\}\,\mathcal{U}\,\{t_e > dl\}\right) \tag{14}$$

We analyzed how network performances vary when there is uncertainty in estimating: 1) the probability of selecting an IP address, and; 2) the probability of message loss during transmission. The former may be biased in a faulty or malicious device. The latter is estimated from empirical data, so it is approximated. Further, the IMDP semantic of IDTMCs (Section 1), which allows a nature to select a different transition distribution at each execution step, properly models the time-varying characteristics of the transmission channel.

In Figure 4, we added uncertainty only to the probability of message loss using the likelihood model, which is suitable for empirically-estimated probabilities. Using classical results from statistics [11], we computed the value of parameter β from Set (2) corresponding to several confidence levels C_L in the measurements. In particular, $0 \leq C_L \leq 1$ and $C_L = 1 - cdf_{\chi_d^2}\left(2 * (\beta_{max} - \beta)\right)$, where $cdf_{\chi_d^2}$ is the cumulative density function of the Chi-squared distribution with d degrees of freedom ($d = 2$ here because there are two possible outcomes, message lost or received). Results show that the value of $P_{s_0}^{max}$ increases by up to $\sim 10\times$ for decreasing C_L, while classical model-checking would only report the value for $C_L = 1$, which roughly over-estimates network performance. The plot can be used by a designer to choose *dl* to make the protocol robust to varying channel conditions, or by a field engineer to assess when the collected measurements are enough to estimate network performances.

In Figure 5, we compose different models of uncertainty, i.e., we also add uncertainty in the probability of selecting the new IP address using the interval model. This probability thus lies in the interval $[(1-u)p_0, (1+u)p_0]$. We, arbitrarily, fixed $dl = 25$ and swept DV_{tot} in the range $[10 - 100]$, which covers most domestic applications, to study how network congestion affects the value of Equation 14. We studied four scenarios: the *ideal* scenario, returned by classical model-checking techniques; the *confident, normal, conservative* scenarios, where we added increasing uncertainty to model different knowledge levels of the network behavior, a

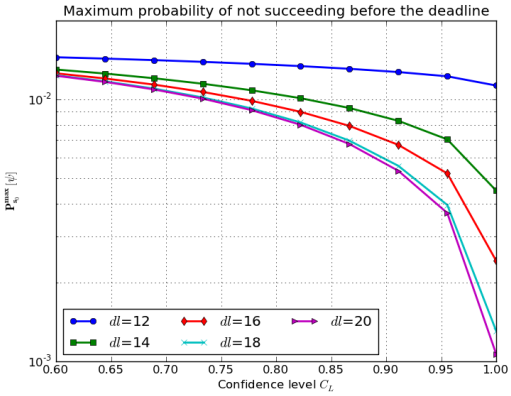

Fig. 4: Value of Equation 14 (top) and verification runtime (bottom).

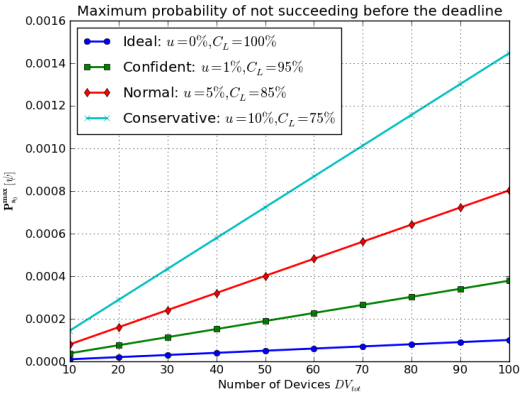

Fig. 5: Value of Eq. 14 for increasing number of devices in the network.

situation that often arises during the different design phases, from conception to deployment. Results show that $P_{s_0}^{max}[\psi]$ gets up to $\sim$ 15$\times$ higher than the ideal scenario, an information that designers can use to determine the most sensitive parameters of the system and to assess the impact of their modeling assumptions on the estimation of network performances.

7 Conclusions and Future Work

We addressed the problem of verifying PCTL properties of Convex-MDPs (CMDPs), i.e., MDPs whose state transition probabilities are only known to lie within convex uncertainty sets. Using results on strong duality for convex programs, we proved that model checking is decidable in PTIME for the fragment of PCTL without the *Bounded Until* operator. For the entire PCTL syntax, the algorithmic complexity becomes pseudo-polynomial in the size of the property. Verification results on two case studies show that uncertainty substantially alters the computed probabilities, thus revealing the importance of the proposed analysis.

As future work, we aim to relax the *rectangular uncertainty* assumption, to limit the power of nature and obtain a less conservative analysis. Also, we plan to verify a complex physical system, e.g. an airplane power system, in which modeling uncertainties are present both in the underlying physical process and in the failure probabilities of its components.

Acknowledgments. The authors thank John B. Finn for the contribution in the first stages of the project and the reviewers for the helpful comments. The research was partially funded by DARPA Award Number HR0011-12-2-0016 and by STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA. Approved for public release; distribution is unlimited. The content of this paper does not necessarily reflect the position or the policy of the US government and no official endorsement should be inferred.

References

1. C. Courcoubetis and M. Yannakakis, "The Complexity of Probabilistic Verification," *Journal of ACM*, vol. 42(4), pp. 857–907, 1995.
2. A. Bianco and L. De Alfaro, "Model Checking of Probabilistic and Nondeterministic Systems," in *Proc. of FSTTCS*, ser. LNCS, vol. 1026, 1995, pp. 499–513.

3. M. Kwiatkowska, "Quantitative Verification: Models, Techniques and Tools," in *Proc. of the* 6^{th} *ACM Special Interest Group on Software Engineering (SIGSOFT)*, 2007, pp. 449–458.
4. H. Hansson and B. Jonsson, "A Logic for Reasoning About Time and Reliability," *Formal Aspects of Computing*, vol. 6(5), pp. 512–535, 1994.
5. M. Kwiatkowska *et al.*, "PRISM 4.0: Verification of Probabilistic Real-Time Systems," *Proc. of* 23^{rd} *Intl. Conf. on Computed Aided Verification*, pp. 585–591, 2011.
6. I. Kozine and L. Utkin, "Interval-Valued Finite Markov Chains," *Reliable Computing*, vol. 8(2), pp. 97–113, 2002.
7. K. Sen *et al.*, "Model-Checking Markov Chains in the Presence of Uncertainties," *Proc. of TACAS*, vol. 3920, pp. 394–410, 2006.
8. K. Chatterjee, K. Sen, and T. Henzinger, "Model-Checking ω-regular Properties of Interval Markov Chains," in *Proc. of FOSSACS*, 2008, pp. 302–317.
9. A. Ben-Tal and A. Nemirovski, "Robust Solutions of Uncertain Linear Programs," *Oper. Res. Lett.*, vol. 25(1), pp. 1–13, 1999.
10. A. Andreychenko *et al.*, "Parameter Identification for Markov Models of Biochemical Reactions," in *Proc. of the* 23^{th} *Intl. Conf. on Computer Aided Verification (CAV)*, 2011, pp. 83–98.
11. A. Nilim and L. El Ghaoui, "Robust Control of Markov Decision Processes with Uncertain Transition Matrices," *Journal of Operations Research*, pp. 780–798, 2005.
12. M. Y. Vardi, "Automatic Verification of Probabilistic Concurrent Finite State Programs," in *Proc. of the 26th Symp. on Foundations of Computer Science*, ser. SFCS, 1985, pp. 327–338.
13. E. Lehmann and G. Casella, *Theory of Point Estimation*. Springer-Verlag, New York, 1998.
14. M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, 1994.
15. V. Forejt *et al.*, "Automated Verification Techniques for Probabilistic Systems," *Formal Methods for Eternal Networked Software Systems (SFM)*, vol. 6659, pp. 53–113, 2011.
16. R. Barbuti *et al.*, "Probabilistic Model Checking of Biological Systems with Uncertain Kinetic Rates," in *Reachability Problems*. Springer Berlin / Heidelberg, 2009, vol. 5797, pp. 64–78.
17. E. M. Hahn *et al.*, "Synthesis for PCTL in Parametric Markov Decision Processes," 2011.
18. M. Kwiatkowska *et al.*, "Automated Verification of a Randomized Distributed Consensus Protocol Using Cadence SMV and PRISM," in *Proc. of CAV*, 2001, pp. 194–206.
19. M. Lahijanian, S. B. Andersson, and C. Belta, "Control of Markov Decision Processes from PCTL Specifications," in *Proc. of the American Control Conference (ACC)*, 2011, pp. 311–316.
20. E. Wolff, U. Topcu, and R. Murray, "Robust Control of Uncertain Markov Decision Processes with Temporal Logic Specifications," *Intl. Conf. on Decision and Control (CDC)*, 2012.
21. A. D'Innocenzo, A. Abate, and J. Katoen, "Robust PCTL Model Checking," in *Proc. of the* 15^{th} *ACM Intl. Conf. on Hybrid Systems: Computation and Control (HSCC)*, 2012, pp. 275–286.
22. E. Clarke and A. Emerson, "Design and Synthesis of Synchronization Skeletons Using Branching Time Temporal Logic," *Proc. of the Workshop on Logic of Programs*, vol. 131, 1981.
23. Y. Nesterov and A. Nemirovski, *Interior-Point Polynomial Algorithms in Convex Programming*, ser. Studies in Applied and Numerical Mathematics, 1994.
24. S. Boyd and L. Vandenberghe, "Convex Optimization," *Cambridge University Press*, 2004.
25. "MOSEK," http://www.mosek.com.
26. Online: http://www.eecs.berkeley.edu/~puggelli/.
27. Online: http://www.prismmodelchecker.org/benchmarks/.
28. J. Aspnes and M. Herlihy, "Fast Randomized Consensus Using Shared Memory," *Journal of Algorithms*, vol. 11(3), pp. 441–461, 1990.
29. S. Cheshire, B. Adoba, and E. Gutterman, "Dynamic configuration of IPv4 link local addresses," available from http://www.ietf.org/rfc/rfc3927.txt.
30. M. Kwiatkowska *et al.*, "Performance Analysis of Probabilistic Timed Automata Using Digital Clocks," *Formal Methods in System Design*, vol. 29, pp. 33–78, 2006.
31. D. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2011.

Appendices

A Convex Optimization Results

In this appendix, we give details on the results from convex theory and duality that we used in the paper. Most of the material is an elaboration of [11].

We begin by giving the definition of a convex set.

Definition A.1. *A set C is convex if the line segment between any two points in C lies in C, i.e., if for any $x, y \in C$ and any α with $0 \leq \alpha \leq 1$, we have: $\alpha x + (1-\alpha)y \in C$ [24].*

The convex sets $\mathcal{F}_s^a, \forall s \in S, \forall a \in \mathcal{A}(s)$ model the uncertainty in the estimation of the rows in the transition matrices of $\mathcal{M}_\mathcal{C}$. In the following, we will also use:

Definition A.2. *A function $h : \mathbb{R}^N \to \mathbb{R}$ is convex if its domain D is a convex set, and for all $\mathbf{x}, \mathbf{y} \in D$ and α with $0 \leq \alpha \leq 1$, we have: $h(\alpha x + (1-\alpha)y) \leq \alpha h(x) + (1-\alpha)h(y)$ [24].*

We now introduce the convex uncertainty models explicitly supported by our framework. In order:

1. Interval
2. Likelihood
3. Ellipsoidal
4. Entropy (not introduced in the paper for space limits)

For each of them, we also give details on the primal and dual formulation of the inner problem:

$$\max_{\mathbf{f}_s^a \in \mathcal{F}_s^a} \mathbf{x}^T \mathbf{f_s^a} \tag{15}$$

and derive the time-complexity of the algorithms used to solve it. Finally, for both the interval and ellipsoidal models of uncertainty, we provide the full formulation of Problem (10) used to verify the $\mathcal{U}$ operator in polynomial time. Similar results can be obtained also for the minimization problem. In the following, we omit state and action indices when possible to improve readability.

A.1 Interval Model

A common description of uncertainty for transition matrices is by intervals:

$$\mathcal{F}_s^a = \{\mathbf{f}_s^a \in \mathbb{R}^N \mid \mathbf{0} \leq \underline{\mathbf{f}}_s^a \leq \mathbf{f}_s^a \leq \overline{\mathbf{f}}_s^a \leq \mathbf{1}, \mathbf{1}^T \mathbf{f}_s^a = 1\} \tag{16}$$

where $\underline{\mathbf{f}}_s^a, \overline{\mathbf{f}}_s^a \in \mathbb{R}^N$ are vectors containing lower and upper bounds of the elements of $\mathbf{f}$. This representation is suitable when the components of the transition matrices are individually estimated by statistical data.

We rewrite the inner problem in Equation (15) in primal form:

$$\begin{aligned} \sigma^*(\mathbf{x}) = \max\ & \mathbf{x}^T \mathbf{f} \\ \text{s.t. } & \mathbf{1}^T \mathbf{f} = 1 \\ & \underline{\mathbf{f}} \leq \mathbf{f} \leq \bar{\mathbf{f}} \end{aligned} \tag{17}$$

The dual problem reads:

$$\begin{aligned} d^*(\mathbf{x}) = \min_{\lambda_1, \boldsymbol{\lambda_2}, \boldsymbol{\lambda_3}}\ & \lambda_1 - \underline{\mathbf{f}}^T \boldsymbol{\lambda_2} + \bar{\mathbf{f}}^T \boldsymbol{\lambda_3} \\ \text{s.t. } & \boldsymbol{\lambda_2} \geq \mathbf{0},\ \boldsymbol{\lambda_3} \geq \mathbf{0} \\ & \mathbf{x} + \boldsymbol{\lambda_2} - \boldsymbol{\lambda_3} - \lambda_1 \mathbf{1} = \mathbf{0} \end{aligned} \tag{18}$$

Since the primal problem is an LP, strong duality holds and $\sigma^* = d^*$ [24].

Replacing Problem (17) with Problem (18), we obtain a new LP formulation for Problem (10), used to verify the $\mathcal{U}$ operator:

$$\begin{aligned}
\min_{\mathbf{x},\lambda^a_{1,s},\boldsymbol{\lambda}^{\mathbf{a}}_{\mathbf{2},\mathbf{s}},\boldsymbol{\lambda}^{\mathbf{a}}_{\mathbf{3},\mathbf{s}}} \quad & \mathbf{x^T 1} \\
\text{s.t. } & x_s = 0 && \forall s \in S^{no} \\
& x_s = 1 && \forall s \in S^{yes} \\
& x_s \geq \lambda^a_{1,s} - (\underline{\mathbf{f}}^s_a)^T \boldsymbol{\lambda}^{\mathbf{a}}_{\mathbf{2},\mathbf{s}} + (\overline{\mathbf{f}}^s_a)^T \boldsymbol{\lambda}^{\mathbf{a}}_{\mathbf{3},\mathbf{s}} && \forall s \in S^?,\ \forall a \in \mathcal{A}(s) \\
& \mathbf{x} + \boldsymbol{\lambda}^{\mathbf{a}}_{\mathbf{2},\mathbf{s}} - \boldsymbol{\lambda}^{\mathbf{a}}_{\mathbf{3},\mathbf{s}} - \lambda^a_{1,s}\mathbf{1} = \mathbf{0} && \forall s \in S^?,\ \forall a \in \mathcal{A}(s) \\
& \boldsymbol{\lambda}^{\mathbf{a}}_{\mathbf{2},\mathbf{s}} \geq \mathbf{0},\ \boldsymbol{\lambda}^{\mathbf{a}}_{\mathbf{3},\mathbf{s}} \geq \mathbf{0} && \forall s \in S^?,\ \forall a \in \mathcal{A}(s)
\end{aligned} \tag{19}$$

During the verification of the *Next* operator instead, we want to solve Problem (18). To derive the time complexity of this operation, we rewrite the problem as [11]:

$$d^* = \min_{\lambda} \left(\bar{\mathbf{f}} - \underline{\mathbf{f}}\right)^T (\lambda \mathbf{1} - \mathbf{x})^+ + \mathbf{x}^T \bar{\mathbf{f}} + \lambda \left(1 - \mathbf{1}^T \bar{\mathbf{f}}\right)$$

where $\mathbf{v}^+$ represent the positive part of vector $\mathbf{v}$. In this form, the dual problem is unconstrained, and it minimizes a convex piecewise function with break-points at the origin and at $x_i,\ i = 1, \cdots N$. A bisection algorithm over the discrete set $b = 0, x_i,\ i = 1, \cdots N$ will thus find the optimal solution in $O\left(Nlog(N)\right)$ steps.

A.2 Likelihood Model

The likelihood model is appropriate when the transition probabilities between states are determined experimentally. The resulting empirical frequencies of transition associated to action $a \in A$ are collected in matrix H^a. Uncertainty in the transition matrices can then be described by the likelihood region [13]:

$$\mathcal{F}^a = \{F^a \in \mathbb{R}^{N \times N} \mid F^a \succeq 0, F^a \mathbf{1} = \mathbf{1}, \textstyle\sum_{s,s'} h^a_{ss'} \log(f^a_{ss'}) \geq \beta^a\}$$

where $\beta^a < \beta^a_{max} = \sum_{s,s'} h^a_{ss'} \log(h^a_{ss'})$ represents the uncertainty level. Since the likelihood region above does not satisfy Assumption 2.1, it must be approximated by projection onto each row of the transition matrix. We obtain:

$$\mathcal{F}^a_s = \{\mathbf{f}^a_s \in \mathbb{R}^N \mid \mathbf{f}^a_s \geq \mathbf{0}, \mathbf{1}^T \mathbf{f}^a_s = 1, \textstyle\sum_{s'} h^a_{ss'} \log(f^a_{ss'}) \geq \beta^a_s\} \tag{20}$$

Even with this approximation, likelihood regions are less conservative uncertainty representations than intervals, which arise from further projection onto the row components.

We rewrite the inner problem in Equation (15) in primal form:

$$\begin{aligned}
\sigma^*(\mathbf{x}) = \max\ & \mathbf{x}^T \mathbf{f}_s \\
\text{s.t. } & \mathbf{1}^T \mathbf{f}_s = 1 \\
& \textstyle\sum_{s'} h_{ss'} \log(f_{ss'}) \geq \beta_s \\
& \mathbf{f}_s \geq \mathbf{0}
\end{aligned} \tag{21}$$

The dual problem reads [11]:

$$\begin{aligned}
d^*(\mathbf{x}) = \min_{\lambda_1,\lambda_2}\ & \lambda_1 - (1+\beta_s)\lambda_2 + \lambda_2 \textstyle\sum_{s'} h_{ss'} \log\left(\frac{\lambda_2 h_{ss'}}{\lambda_1 - x_{s'}}\right) \\
\text{s.t. } & \lambda_1 \geq x_{max} = \max_{s' \in S} x_{s'} \\
& \lambda_2 \geq 0
\end{aligned} \tag{22}$$

The primal problem is convex, and it satisfies Slater's condition [24] for non-trivial uncertainty sets, i.e. for $\beta_s < \beta_{max} = \sum_{s,s'} h_{ss'} \log(h_{ss'})$, so strong duality holds and $\sigma^* = d^*$. Also, it can be proven that the dual Problem (22) is jointly convex in λ and $\mathbf{x}$.

Replacing Problem (21) with Problem (22), we thus obtain a new formulation for Problem (10), used to verify the $\mathcal{U}$ operator:

$$\begin{aligned}
&\min_{x_s, \lambda^a_{1,s}, \lambda^a_{2,s}} \ \mathbf{x_s^T 1} \\
&\quad \lambda^{\mathbf{a}}_{\mathbf{3,s}}, \lambda^{\mathbf{a}}_{\mathbf{4,s}} \\
&\quad \text{s.t. } x_s = 0 && \forall s \in S^{no} \\
&\qquad x_s = 1 && \forall s \in S^{yes} \\
&\qquad x_s \geq \lambda^a_{1,s} - (1+\beta^a_s)\lambda^a_{2,s} + \lambda^a_{2,s} \textstyle\sum_{s'} h^a_{ss'} \log\left(\frac{\lambda^a_{2,s} h^a_{ss'}}{\lambda^a_{1,s} - x_{s'}}\right) && \forall s \in S^?, \forall a \in A \\
&\qquad \lambda^a_{1,s} \geq x_{max} = \max_{s' \in S} x_{s'} && \forall s \in S^?, \forall a \in A \\
&\qquad \lambda^a_{2,s} \geq 0 && \forall s \in S^?, \forall a \in A
\end{aligned} \tag{23}$$

Moreover, when verifying the *Next* operator, the dual problem can be reduced to one dimension and solved using a bisection algorithm [11], with resulting time complexity $O\left(Nlog(x_{max}/\epsilon)\right))$ [24] with ϵ equal to the machine precision and $x_{max} \leq 1$, since $\mathbf{x}$ is a vector of probabilities.

A.3 Ellipsoidal Model

Ellipsoidal models can be seen as a second-order approximation of the likelihood model [11]. Intuitively, in this model the elements of $\mathbf{f}^a_s \in \mathcal{F}^a_s$ are restricted to lie on the intersection of the ellipse $\mathbf{E}^a_s = \{\mathbf{f}^a_s \mid \|R^a_s \mathbf{f}^a_s\|_2 \leq 1, R^a_s \succ 0\}$ and the probability simplex $\Delta_N = \{\mathbf{f}^a_s \in \mathbb{R}^N \mid \mathbf{1}^T \mathbf{f}^a_s = 1, \mathbf{f}^a_s \geq \mathbf{0}\}$. We will thus consider sets:

$$\mathcal{F}^a_s = \{\mathbf{f}^a_s \in \mathbb{R}^N \mid \mathbf{f}^a_s \geq \mathbf{0}, \mathbf{1}^T \mathbf{f}^a_s = 1, \|R^a_s \left(\mathbf{f}^a_s - \mathbf{h}^a_s\right)\|_2 \leq 1, R^a_s \succ 0\} \tag{24}$$

where the matrix R^a_s represents an ellipsoidal approximation of the likelihood region $r^a_s = \{\mathbf{f}^a_s \in \mathbb{R}^N \mid \sum_{s'} h^a_{ss'} \log(f^a_{ss'}) \geq \beta^a_s\}$. In particular, R^a_s can be computed as follows. First, we write the second order approximation of the likelihood region:

$$r^a_s = \{\mathbf{f}^a_s \in \mathbb{R}^N \mid \textstyle\sum_{s'} h^a_{ss'} \log(f^a_{ss'}) \geq \beta^a_s\} \approx \{\mathbf{f}^a_s \in \mathbb{R}^N \mid \sum_{s'} \frac{\left(f^a_{ss'} - h^a_{ss'}\right)^2}{h^a_{ss'}} \leq \mathcal{K}^2\} \tag{25}$$

where $\mathcal{K}^2 = 2\left(\beta_{max} - \beta\right)$ is a measure of the uncertainty in approximating the values $h^a_{ss'}$. The approximation in Equation 25 can be written in matrix form:

$$\textstyle\sum_{s'} \frac{\left(f^a_{ss'} - h^a_{ss'}\right)^2}{h^a_{ss'}} \leq \mathcal{K}^2 \Leftrightarrow \|R^a_s \left(\mathbf{f}^a_s - \mathbf{h}^a_s\right)\|_2 \leq 1 \tag{26}$$

$$R^a_s = \begin{bmatrix} (\sqrt{h^a_{ss_0}}\mathcal{K})^{-1} & 0 & \cdots & 0 \\ 0 & (\sqrt{h^a_{ss_1}}\mathcal{K})^{-1} & \cdots & 0 \\ & & & \\ 0 & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & (\sqrt{h^a_{ss_N}}\mathcal{K})^{-1} \end{bmatrix}$$

Matrix R^a_s is guaranteed to be positive definite, i.e., $R^a_s \succ 0$, because it is diagonal and the values $h^a_{ss'}$ represent empirical frequencies of transition from states s to s', hence they are non-negative by definition.

We rewrite the inner problem in Equation (15) in primal form:

$$\begin{aligned} \sigma^*(\mathbf{x}) = \max_{\mathbf{f}_s^a} \ & \mathbf{x}^T \mathbf{f}_s^a \\ \text{s.t. } & \mathbf{1}^T \mathbf{f}_s^a = 1 \\ & \| R_s^a \left(\mathbf{f}_s^a - \mathbf{h}_s^a \right) \|_2 \leq 1 \\ & \mathbf{f}_s^a \geq \mathbf{0} \end{aligned} \tag{27}$$

The dual reads:

$$\begin{aligned} d^*(\mathbf{x}) = \min_{\lambda_1, \lambda_2, \boldsymbol{\lambda_3}} \ & \lambda_1 + \lambda_2 + \mathbf{h}^T R \boldsymbol{\lambda_3} \\ \text{s.t. } & \| \boldsymbol{\lambda_3} \|_2 \leq \lambda_2 \\ & \mathbf{x} - \lambda_1 \mathbf{1} - R^T \boldsymbol{\lambda_3} = \mathbf{0} \\ & \lambda_2 \geq 0, \quad \boldsymbol{\lambda_3} \geq \mathbf{0} \end{aligned} \tag{28}$$

where the state and action indices have been dropped to improve readability. The inner problem is a Second Order Cone Problem (SOCP), which satisfies Slater's condition [24] for any non-trivial uncertainty set, so strong duality holds and $\sigma^* = d^*$. The dual Problem (28) is an SOCP, so it is trivially jointly convex in λ and $\mathbf{x}$.

Replacing Problem (27) with Problem (28), we can thus obtain a new SOCP formulation for Problem (10), used to verify the $\mathcal{U}$ operator:

$$\begin{aligned} \min_{x_s, \lambda_{1,s}^a, \lambda_{2,s}^a, \boldsymbol{\lambda}_{\mathbf{3,s}}^{\mathbf{a}}} \ & \mathbf{x_s^T 1} & & \\ \text{s.t. } & x_s = 0 & & \forall s \in S^{no} \\ & x_s = 1 & & \forall s \in S^{yes} \\ & x_s \geq \lambda_{1,s}^a + \lambda_{2,s}^a + \mathbf{h_s^{a^T}} R_s^a \boldsymbol{\lambda}_{\mathbf{3,s}}^{\mathbf{a}} & & \forall s \in S^?, \forall a \in A \\ & \| \boldsymbol{\lambda}_{\mathbf{3,s}}^{\mathbf{a}} \|_2 \leq \lambda_{2,s}^a & & \forall s \in S^?, \forall a \in A \\ & \mathbf{x} - \lambda_{1,s}^a \mathbf{1} - R_s^{a^T} \boldsymbol{\lambda}_{\mathbf{3,s}}^{\mathbf{a}} = \mathbf{0} & & \forall s \in S^?, \forall a \in A \\ & \lambda_{2,s}^a \geq 0, \quad \boldsymbol{\lambda}_{\mathbf{3,s}}^{\mathbf{a}} \geq \mathbf{0} & & \forall s \in S^?, \forall a \in A \end{aligned} \tag{29}$$

Introducing uncertainty comes at the cost of solving an SOCP with $(N+2)MQ$ more variables and $(N+1)MQ$ more constraints than the original LP, where $Q = |S^?| = O(N)$.

During the verification of the *Next* operator instead, we want to solve either Problem (27) or Problem (28). Since they are both SOCP, they can be solved using interior-point methods with worst-case (*practical*) time complexity $O\left(N^{1.5} log(x_{max} \setminus \epsilon)\right)$ $\left(O\left(N log(x_{max} \setminus \epsilon)\right)\right)$ [24] and $x_{max} \leq 1$, since $\mathbf{x}$ is a vector of probabilities.

Finally, we report a second dual formulation of Equation (15), which we experimentally found to have better runtime performance when using the VI routine to verify properties containing the $\mathcal{U}$ operator. Intuitively, the faster performance is achieved because this formulation allows to write the analytical expression of the dual solution of the inner problem as a function of the state probabilities $\mathbf{x}$. The analytical expression can then be rapidly evaluated during the VI iterations using the values of $\mathbf{x}$ estimated in the previous iteration. We note that the same approach cannot be used for the CP routine, since the derived analytical expression is not easily representable in conic form when the optimization Problem (29) operates both on the decision variables $\mathbf{x}$ and λ at the same time. In the following, we will drop the state and action indices to improve readability.

We rewrite the inner problem in Equation (27) in an equivalent primal form:

$$\begin{aligned} \sigma^*(\mathbf{x}) = \max_{\mathbf{f}} \ & \mathbf{x}^T\mathbf{f} \\ \text{s.t. } & \mathbf{1}^T\mathbf{f} = 1 \\ & \sum_{s'} \frac{(f_{s'} - h_{s'})^2}{h_{s'}} \leq \mathcal{K}^2 \\ & \mathbf{f} \geq \mathbf{0} \end{aligned} \tag{30}$$

The Lagrangian operator associated to Problem (30) reads:

$$\mathcal{L}(\mathbf{f}, \mu, \xi, \nu) = \mathbf{x}^T\mathbf{f} + \mu(1 - \mathbf{1}^T\mathbf{f}) + \xi^T\mathbf{f} + \nu\left(\mathcal{K}^2 - \sum_{s'} \frac{(f_{s'} - h_{s'})^2}{h_{s'}}\right) \tag{31}$$

The primal optimal value can be computed as:

$$\sigma^*(\mathbf{x}) = \max_{\mathbf{f}} \min_{\mu,\xi,\nu} \mathcal{L}(\mathbf{f}, \mu, \xi, \nu) \tag{32}$$

By the *minimax* theorem [24], we obtain an upper bound on the value of $\sigma^*(\mathbf{x})$ by inverting the "max" and "min" operators:

$$d^*(\mathbf{x}) = \min_{\mu,\xi,\nu} \max_{\mathbf{f}} \mathcal{L}(\mathbf{f}, \mu, \xi, \nu) \tag{33}$$

Problem (30) satisfies Slater's condition [24] for any non-trivial uncertainty set, so strong duality holds and $\sigma^* = d^*$. We can thus solve Problem (33) instead of Problem (30) while preserving the soundness and completeness of the verification procedure. As a first step, we compute the dual function $g(\mu, \xi, \nu)$ by solving the inner problem in Equation (33), i.e., we aim at computing:

$$g(\mu, \xi, \nu) = \max_{\mathbf{f}} \mathcal{L}(\mathbf{f}) \tag{34}$$

We can solve Problem (34) by setting the gradient of the Lagrangian to zero, and solving for the optimal primal solution $\mathbf{f}^*$:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial f_{s_0}} = x_0 - \mu + \xi_0 - \frac{2\nu}{h_{s_0}}(f_{s_0} - h_{s_0}) = 0 & f^*_{s_0} = \frac{h_{s_0}}{2\nu}(x_0 - \mu + \xi_0) + h_{s_0} \\ \frac{\partial \mathcal{L}}{\partial f_{s_1}} = x_1 - \mu + \xi_1 - \frac{2\nu}{h_{s_1}}(f_{s_1} - h_{s_1}) = 0 \quad \Rightarrow & f^*_{s_1} = \frac{h_{s_1}}{2\nu}(x_1 - \mu + \xi_1) + h_{s_1} \\ \quad \dots & \dots \\ \frac{\partial \mathcal{L}}{\partial f_{s_N}} = x_N - \mu + \xi_N - \frac{2\nu}{h_{s_N}}(f_{s_N} - h_{s_N}) = 0 & f^*_{s_N} = \frac{h_{s_N}}{2\nu}(x_N - \mu + \xi_N) + h_{s_N} \end{cases} \tag{35}$$

Substituting $\mathbf{f}^*$ back into Problem 34, we obtain the dual function:

$$g(\mu, \xi \geq 0, \nu \geq 0) = \mu + \nu\mathcal{K}^2 + \sum_{s'}(h_{s'}(x_{s'} - \mu + \xi_{s'})) + \frac{1}{4\nu}\sum_{s'}(h_{s'}(x_{s'} - \mu + \xi_{s'})^2) \tag{36}$$

We can now compute d^* solving the dual problem:

$$d^* = \min_{\mu,\xi\geq 0,\nu\geq 0} g(\mu, \xi, \nu) \tag{37}$$

Before further proceeding, we note that, for monotonicity reasons, we can trivially set $\xi^* = 0$. We thus aim to solve the following optimization problem:

$$d^* = \min_{\mu,\nu \geq 0} \mu + \nu\mathcal{K}^2 + \sum_{s'}(h_{s'}(x_{s'} - \mu)) + \frac{1}{4\nu}\sum_{s'}(h_{s'}(x_{s'} - \mu)^2) \tag{38}$$

We first set the partial derivatives of the dual function g to zero and compute the optimal dual solution (μ^*, ν^*). Formally:

$$\begin{cases} \frac{\partial g}{\partial \mu} = 1 - \frac{2\mathbf{h}}{4\nu}(\mathbf{x} - \mu\mathbf{1})^T - \mathbf{1}^T\mathbf{h} = 0 & \\ \frac{\partial g}{\partial \nu} = \mathcal{K}^2 - \frac{\mathbf{h}(\mathbf{x}-\mu\mathbf{1})^{2T}}{4\nu^2} = 0 & \end{cases} \Rightarrow \begin{array}{l} \mu^* = \sum_{s'} h_{s'} x_{s'} \\ \\ \nu^* = \frac{1}{2\mathcal{K}}\sqrt{\sum_{s'} h_{s'}(x_{s'} - \mu)^2} \end{array} \tag{39}$$

The optimal value can then be computed as $d^* = g(\mu^*, \nu^*)$.

Finally, we report the primal and dual formulations to solve the inner problem when verifying properties of the form $P_{\triangleright p}[\psi]$. In the primal problem of Equation (15), we change the optimization operator from "max" to "min":

$$\begin{aligned} \sigma^*(\mathbf{x}) = \min_{\mathbf{f}}\ & \mathbf{x}^T\mathbf{f} \\ \text{s.t. } & \mathbf{1}^T\mathbf{f}_s^a = 1 \\ & \sum_{s'} \frac{(f_{s'} - h_{s'})^2}{h_{s'}} \leq \mathcal{K}^2 \\ & \mathbf{f} \geq \mathbf{0} \end{aligned} \tag{40}$$

Following the same steps presented above, we obtain the corresponding dual problem:

$$d^* = \max_{\mu,\nu \geq 0} \mu - \nu\mathcal{K}^2 + \sum_{s'}(h_{s'}(x_{s'} - \mu)) - \frac{1}{4\nu}\sum_{s'}(h_{s'}(x_{s'} - \mu)^2) \tag{41}$$

which admits the optimal solution:

$$\begin{cases} \frac{\partial g}{\partial \mu} = 1 + \frac{2\mathbf{h}}{4\nu}(\mathbf{x} - \mu\mathbf{1})^T - \mathbf{1}^T\mathbf{h} = 0 & \\ \frac{\partial g}{\partial \nu} = -\mathcal{K}^2 + \frac{\mathbf{h}(\mathbf{x}-\mu\mathbf{1})^{2T}}{4\nu^2} = 0 & \end{cases} \Rightarrow \begin{array}{l} \mu^* = \sum_{s'} h_{s'} x_{s'} \\ \\ \nu^* = \frac{1}{2\mathcal{K}}\sqrt{\sum_{s'} h_{s'}(x_{s'} - \mu)^2} \end{array} \tag{42}$$

A.4 Entropy Model

The entropy model of uncertainty can be viewed as a variation of the likelihood model. In the likelihood setting we bound the divergence from an empirically extracted distribution, whereas in the entropy setting we bound the divergence from a reference analytical distribution q [11]. We will thus consider sets:

$$\mathcal{F}_s^a = \{\mathbf{f}_s^a \in \mathbb{R}^N \mid \mathbf{f}_s^a \geq \mathbf{0}, \mathbf{1}^T\mathbf{f}_s^a = 1, \textstyle\sum_{s'} f_{ss'}^a \log\left(\frac{f_{ss'}^a}{q_{ss'}^a}\right) \leq \beta_s^a\} \tag{43}$$

We rewrite the inner problem in Equation (15) in primal form:

$$\begin{aligned} \sigma^* = \max\ & \mathbf{x}^T\mathbf{f}_s \\ \text{s.t. } & \mathbf{1}^T\mathbf{f}_s = 1 \\ & \textstyle\sum_{s'} f_{ss'} \log\left(\frac{f_{ss'}}{q_{ss'}}\right) \leq \beta_s \\ & \mathbf{f}_s \geq \mathbf{0} \end{aligned} \tag{44}$$

The dual problem reads:

$$d^* = \min_{\lambda} \lambda \log\left(\sum_{s'} q_{ss'} \exp\left(\frac{x_{s'}}{\lambda}\right)\right) + \beta_s \lambda \tag{45}$$
$$\text{s.t. } \lambda \geq 0$$

The primal problem is convex, and it satisfies Slater's condition [24] for non-trivial uncertainty sets, i.e. for $\beta_s > 0$, so strong duality holds and $\sigma^* = d^*$.

Replacing Problem (44) with Problem (45), we can thus obtain a new formulation for Problem (10), used to verify the $\mathcal{U}$ operator:

$$\min_{x_s, \lambda_s^a} \mathbf{x_s^T 1}$$
$$\text{s.t. } x_s = 0 \qquad \forall s \in S^{no} \tag{46a}$$
$$x_s = 1 \qquad \forall s \in S^{yes} \tag{46b}$$
$$x_s \geq \lambda_s^a \log\left(\sum_{s'} q_{ss'} \exp\left(\frac{x_{s'}}{\lambda_s^a}\right)\right) + \beta_s \lambda_s^a \qquad \forall s \in S^?, \forall a \in A \tag{46c}$$
$$\lambda_s^a \geq 0 \qquad \forall s \in S^?, \forall a \in A \tag{46d}$$

We prove its joint convexity in $\mathbf{x}$ and λ_s^a as follows. The cost function and Constraints (46a), (46b) and (46d) are trivially convex. Constraint(46c) is generated by a primal-dual transformation, so, according to convex theory, it is convex in the dual variables λ_s^a by construction. Moreover, we prove that it is also jointly convex in $\mathbf{x}$ by induction on the number NS of next states $s' \in S$ for state s. As a base case, $NS = 1$, and we can rewrite the constraint as:

$$x_s \geq \lambda_s^a \log\left(q_{ss'}\right) + x_{s'} + \beta_s \lambda_s^a \tag{47}$$

which is trivially jointly convex. We now assume that the constraint is jointly convex for $NS = n$, and prove that it is jointly convex also for $NS = n + 1$. This result immediately follows from observing that increasing NS simply introduces one more term in the summation, so if the constraint is jointly convex for $NS = n$ it must remain jointly convex also for $NS = n + 1$, since an affine addition preserves convexity according to convex theory. We conclude that Problem (46) is convex.

Moreover, when verifying the *Next* operator, the dual problem is unidimensional and it can thus be solved using a bisection algorithm [11], with resulting time complexity $O\left(Nlog(x_{max} \setminus \epsilon)\right)$ [24] with ϵ equal to the machine precision and $x_{max} \leq 1$, since $\mathbf{x}$ is a vector of probabilities.

B Proof of Contraction Lemma

In this appendix, we present a verification procedure for the Until operator based on Value Iteration (VI), which should be considered as an alternative to the CP procedure. We report also the VI procedure because it has shorter runtime than the CP procedure for some of the analyzed case studies, depending on the structure and data of the problem. Both procedures should be run in the early stages of system verification and the one which performs better should be used in the rest of the project development.

We start by defining:

Definition B.1. *Contraction. Let (B, d) be a metric space and $g : B \to B$. Function g is a* contraction *if there is a real number θ, $0 \leq \theta < 1$, such that:*

$$d\left(g(u), g(v)\right) \leq \theta d(u, v) \quad \forall u, v \in B \tag{48}$$

In the following, we will use:

Proposition B.1. *Contraction mapping. Let (B, d) be a complete metric space and $g : B \to B$ a contraction. Then there exists a unique point $x^* \in B$ such that:*

$$g(x^*) = x^*$$

Additionally, if $x \in B$, then:

$$\lim_{k \to +\infty} g^k(x) = x^*$$

We use the mapping $g = G$ defined as:

$$G = \begin{cases} 0 & \forall s \in S^{no} \\ 1 & \forall s \in S^{yes} \\ 0 & \forall s \in S^{?} \wedge i = 0 \\ \max\limits_{a \in \mathcal{A}(s)} \max\limits_{\mathbf{f}_s^a \in \mathcal{F}_s^a} (\mathbf{x}^{i-1})^T \mathbf{f}_\mathbf{s}^\mathbf{a} & \forall s \in S^{?} \wedge i \geq 0 \end{cases} \tag{49}$$

where $S^{yes} \stackrel{def}{=} Sat\left(P_{\geq 1}[\phi_1 \mathcal{U} \phi_2]\right)$, $S^{no} \stackrel{def}{=} Sat\left(P_{\leq 0}[\phi_1 \mathcal{U} \phi_2]\right)$ and $S^{?} = S \setminus (S^{no} \cup S^{yes})$. We note that MQ convex problems need to be solved to compute mapping G, with $Q = \mid S^{?} \mid$. For the uncertainty models considered in the paper, each problem can be solved with complexity $O\left(N \log(1/\epsilon)\right)$ [11], and ϵ equal to the machine precision. To simplify notation in the proof, we use the weighted maximum norm $\| \, . \, \|_\mathbf{w}$ of a vector $v \in \mathbb{R}^N$ defined as:

$$\| \, . \, \|_\mathbf{w} = \max_{i=1 \cdots N} \frac{\mid v_i \mid}{w_i} \tag{50}$$

where w_i is the scalar weight associated to each element of v.

We can now state:

Lemma B.1. *Mapping G is a contraction over the metric space $(\mathbb{R}^N, \| \, . \, \|_\mathbf{w})$.*

Proof. The proof follows closely the ones in [B.1] (Vol. II, Section 2.4) and [20]. Those proofs refer to a control setting, where the optimal action (control) can be selected. Hence, the contraction needs to hold for only one of the available actions, i.e. the optimal one (existential quantification). Conversely, in the verification setting, the contraction needs to hold across all available actions, because we consider the worst case resolution of nondeterminism (universal

quantification). Further, as in [20], we quantify across all nature behaviors: this is possible due to Assumption 2.2. For the sake of brevity, in the following we will only consider the calculation of Pr_s^{min}, but the same reasoning applies also for the maximization problem.

We start from partitioning the state space $S = S^{yes} \cup S^{no} \cup S^{?}$ as explained in Section 5.2. Since at all iterations the probabilities Pr_s^{min} will remain constant by construction in all states $s \in S^{yes} \cup S^{no}$, we do not need to consider these states explicitly. In particular, we perform the following transformations of the CMDP underlying graph: we collapse the set S^{yes} into one terminal state t, and eliminate all states $s \in S^{no}$ from the graph. These transformations are fundamental together with Assumption 2.2 to guarantee that all possible adversaries *Adv* are *proper* in the transformed graph, i.e. they almost surely reach the terminal state t for all transition matrices in $\mathcal{F}$ [31]. We will now work with the new state space $S^{\dagger} = S^{?} \cup \{t\}$, and, for simplicity, we redefine $N = \mid S^{\dagger} \mid$. We further partition $S^{\dagger}$, as follows. Let $S_1 = \{t\}$ and for $q = 2, 3, \cdots$ compute:

$$S_q = \{s \in S^{\dagger} \mid s \notin S_1 \cup \cdots \cup S_{q-1}, \min_{a \in \mathcal{A}(s)} \max_{s' \in S_1 \cup \cdots \cup S_{q-1}} \min_{f_s^a \in \mathcal{F}_s^a} f_{ss'}^a > 0\}$$

Let r be the largest integer such that S_r is nonempty. Since all adversaries are *proper*, we are guaranteed that $\cup_{q=1}^{r} S_q = S^{\dagger}$. We now need to choose weights $w_s, \forall s \in S^{\dagger}$ such that G is a contraction with respect to $\|.\|_{\mathbf{w}}$. First, we take the s^{th} component w_s to be the same for states s in the same set S_q. Then we set $w_s = y_q$ if $i \in S_q$, where $y_1, \cdots, y_r$ are scalars satisfying $1 = y_1 < y_2 < \cdots < y_r$. Further, let:

$$\xi = \min_{q=2,\cdots,r} \min_{a \in A} \min_{s \in S_q} \min_{\mathbf{f}_s^a \in \mathcal{F}_s^a} \sum_{s' \in S_1 \cup \cdots \cup S_{q-1}} \mathbf{f}_{ss'}^a$$

By construction $0 < \xi \leq 1$.

The rest of the proof goes as follows: first, we will show that if we can find $y_2, \cdots, y_r$ such that for $q = 2, \cdots, r$:

$$\frac{y_r}{y_q}(1 - \xi) + \frac{y_{q-1}}{y_q} \leq \theta$$

for some $\theta < 1$, then G is a contraction. Second, we will prove that such values always exist. We begin by defining:

$$G_s(\mathbf{x}) = \min_{a \in \mathcal{A}(s)} \min_{\mathbf{f}_s^a \in \mathcal{F}_s^a} \mathbf{x}^T \mathbf{f}_s^a$$
$$G_s^a(\mathbf{x}) = \min_{\mathbf{f}_s^a \in \mathcal{F}_s^a} \mathbf{x}^T \mathbf{f}_s^a$$

i.e. the s^{th} element of the output of mapping G applied to vector $\mathbf{x} \in \mathbb{R}^N$, and the same element when mapping G is evaluated only at the fixed action $a \in \mathcal{A}(s)$. Then, for all vectors $\mathbf{v}, \mathbf{u} \in \mathbb{R}^N$, we determine $\mathcal{A}(s)$ such that:

$$a = \underset{\mathcal{A}(s)}{\operatorname{argmin}}\, G(u)$$

We can thus write for all $s \in S^{\dagger}$:

$$\begin{aligned} G_s(\mathbf{v}) - G_s(\mathbf{u}) &= G_s(\mathbf{v}) - G_s^a(\mathbf{u}) \\ &\leq G_s^a(\mathbf{v}) - G_s^a(\mathbf{u}) \\ &= \textstyle\sum_{s'} \left(V_{ss'}^a v_{s'} - U_{ss'}^a u_{s'}\right) \\ &\leq \textstyle\sum_{s'} M_{ss'}^a \left(v_{s'} - u_{s'}\right) \end{aligned}$$

where:

$$\mathbf{V}_s^a = \underset{\mathbf{f}_s^a \in \mathcal{F}_s^a}{\operatorname{argmin}}\ \mathbf{v}^T \mathbf{f}_s^a$$
$$\mathbf{U}_s^a = \underset{\mathbf{f}_s^a \in \mathcal{F}_s^a}{\operatorname{argmin}}\ \mathbf{u}^T \mathbf{f}_s^a$$
$$M_{ss'}^a = \operatorname{argmax}\left\{V_{ss'}^a\left(v_{s'} - u_{s'}\right), U_{ss'}^a\left(v_{s'} - u_{s'}\right)\right\}$$

Let $q(s)$ be such that state s belongs to the set $S_{q(s)}$. Then, for any constant c:

$$\|\mathbf{v} - \mathbf{u}\|_{\mathbf{w}} \Rightarrow v_s - u_s \leq c y_{q(s)} \quad \forall s \in S^\dagger$$

We can thus write $\forall s \in S_q$ and $q = 1, \cdots, r$:

$$\begin{aligned}
\tfrac{G_s(v) - G_s(u)}{c y_{q(s)}} \leq \tfrac{1}{y_{q(s)}} \textstyle\sum_{s' \in S^\dagger} M_{ss'}^a y_{q(s')} & \\
&\leq \tfrac{y_{q(s)-1}}{y_{q(s)}} \textstyle\sum_{s' \in S_1 \cup \cdots \cup S_{q(s)-1}} M_{ss'}^a \\
&+ \tfrac{y_r}{y_{q(s)}} \textstyle\sum_{s' \in S_{q(s)} \cup \cdots \cup S_r} M_{ss'}^a \\
&= \left(\tfrac{y_{q(s)-1}}{y_{q(s)}} - \tfrac{y_r}{y_{q(s)}}\right) \textstyle\sum_{s' \in S_1 \cup \cdots \cup S_{q(s)-1}} M_{ss'}^a \\
&+ \frac{y_r}{y_{q(s)}} \leq \left(\frac{y_{q(s)-1}}{y_{q(s)}} - \frac{y_r}{y_{q(s)}}\right) \xi + \frac{y_r}{y_{q(s)}} \leq \theta
\end{aligned}$$

We have thus proved that $\frac{G_s(v) - G_s(u)}{w_i} \leq c\theta$, for an arbitrary state $s \in S^\dagger$. Taking the maximum over $S^\dagger$, we get:

$$\|G(v) - G(u)\|_{\mathbf{w}} \leq c\theta,\ \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^N \text{s.t.} \|\mathbf{v} - \mathbf{u}\| \leq c$$

so, G is a contraction over the metric space $(\mathbb{R}^N, \|\ .\ \|_{\mathbf{w}})$, and:

$$\theta = \max_{1 \leq q \leq r} \frac{y_r}{y_q}(1 - \xi) + \frac{y_{q-1}}{y_q} \tag{51}$$

is the corresponding contraction factor. Finally, we constructively prove by induction that it is always possible to find scalars $y_1, \cdots, y_r$ such that the above assumptions hold. As the base case, we set $y_0 = 0, y_1 = 1$. At the induction step, we suppose that $y_2, \cdots, y_q$ have already been determined. If $\xi = 1$, we set $y_{q+1} = y_q + 1$. If $\xi < 1$, we set $y_{q+1} = \frac{1}{2}\left(y_q + m_q\right)$ where:

$$m_q = \min_{1 \leq i \leq q} \left\{ y_i + \frac{\xi}{1 - \xi}(y_i - y_{i-1}) \right\}$$

With these choices, we are guaranteed that:

$$m_{q+1} = \min\ \left\{ m_q, y_i + \frac{\xi}{1 - \xi}(y_i - y_{i-1}) \right\}$$

so by induction, we have that $y_q < y_{q+1} < m_{q+1}$, and we can construct the required sequence. □

We now state the main results of this appendix:

Lemma B.2. *The VI procedure to verify the* Unbounded Until *operator is sound and complete, i.e.,*

$$\mathbf{P}^{max}[\phi_1 \mathcal{U} \phi_2] = \lim_{k \to +\infty} G^k(\mathbf{x}) \tag{52}$$

In practice, we need a criterion to stop the iteration, so an error is introduced in Equation (52) and P^{max} is approximated. In the following, we show how to compute an exact lower bound K^{min} on the number of iterations required to obtain the desired accuracy ϵ_d. The existence of such a lower bound also proves the soundness and completeness of the VI procedure. Further, we propose a heuristic stopping criterion based on a relative tolerance check, which is often used to reduce runtime [5]. The user can choose between the two approaches, depending on the application.

Exact Lower Bound. From Lemma B.1, at the end of the i^{th} iteration the residual error in estimation is bounded by:

$$\rho_i = \|\mathbf{P}^{max} - \mathbf{x}^i\|_{\mathbf{w}} \leq \rho_0 \frac{\theta^i}{1-\theta}$$

where ρ_0 is the initial error in estimation which can be trivially bounded by $\rho_0 \leq 1$. The error in the estimation of the satisfaction probabilities is bounded by ϵ_d if:

$$\rho_i \leq \frac{\epsilon_d}{w_{max}} \Rightarrow \mid P_s^{max} - x^i \mid \leq \epsilon_d, \;\; \forall s \in S^?$$

where w_{max} is the maximum of the weights of Norm (50). We can thus obtain a lower bound K^{min} for the number of iterations required to achieve the desired accuracy ϵ_d:

$$\frac{\theta^K}{1-\theta} \leq \frac{\epsilon_d}{w_{max}} \rightarrow K \geq K^{min} = \frac{log[\epsilon_d(1-\theta)] - log(w_{max})}{log(\theta)} \tag{53}$$

Heuristic Based on Relative Tolerance. Although provably exact, the lower bound K^{min} in the number of iterations derived in the previous section might be too conservative and result in an unnecessary increase in runtime. We thus also present a heuristic stopping criterion based on relative tolerance. In particular, we stop when:

$$\delta_r > \max_{s \in S^?} \left(|x_s^i - x_s^{i-1}| / x_s^i \right) \tag{54}$$

the maximum relative difference in the computed value of $P_s^{max}[\phi_1 \mathcal{U} \phi_2], \; \forall s \in S^?$ between two consecutive iterations is below a user defined tolerance δ_{rel}. We note that Such a criterion does not guarantee that the error is bounded by δ_r. The required δ_r to achieve accuracy ϵ_d, i.e., δ_r^*, depends on the CMDP model and needs to be determined by trial-and-error, a common practice in iterative procedures (e.g. the ODE solver in a circuit simulator). To determine δ_r^*, we compute several approximations of $P^{max}[\psi]$ while decreasing δ_r by steps of $10\times$. We heuristically stop when no probability $P_s^{max}, \forall s \in S^?$, changes more than ϵ_d after checking Criterion (54) for δ_r^* and $\delta_r^*/100$. Finally, errors in solving the inner problems, as introduced in Section 5.3, are propagated across iterations. We call ϵ_{inn} the inner problem accuracy. If the VI procedure exits after I iterations and $\epsilon_d < I \times \epsilon_{inn}$, the procedure needs to be run again after decreasing ϵ_{inn} to, approximately, $\epsilon_{inn} < \epsilon_d / I$.

We use the VI routine with $\epsilon_d = 10^{-3}$ to verify again $\phi = P_{\geq 0.3}[\, \vartheta \, \mathcal{U} \, \omega \,]$ in the example in Figure 1. After 3 iterations, we get $\mathbf{P}^{min}[\, \vartheta \, \mathcal{U} \, \omega \,] = [0.2, 0, 1, 0.32]$ and $Sat(\phi) = \{s_2, s_3\}$.

[B.1] D. Bertsekas, *"Dynamic Programming and Optimal Control"*, Athena Scientific, 2011

C Toy LP

This appendix reports the full LP formulation that was used to verify property $\phi = P_{\geq 0.3}[\ \vartheta\ \mathcal{U}\ \omega\ ]$ on the example in Figure 1. Problem (10) written with the data of the model has 19 variables and 11 constraints. All variables are (implicitly) bounded to be positive apart from the ones labeled as *free* at the bottom of the formulation.

$$\max_{\mathbf{x},\lambda_1,\boldsymbol{\lambda_2},\boldsymbol{\lambda_3}} \; x_0 + x_3 \tag{55}$$

Subject to:

$$\begin{aligned}
& x_2 = 1 \\
& x_1 = 0 \\
& x_0 \leq \lambda^a_{1,s_0} + 0.6\lambda^a_{2,s_0s_0} + 0.2\lambda^a_{2,s_0s_1} - 0.8\lambda^a_{3,s_0s_0} - 0.5\lambda^a_{3,s_0s_1} \\
& x_1 - \lambda^a_{1,s_0} + \lambda^a_{3,s_0s_0} - \lambda^a_{2,s_0s_0} = 0 \\
& x_2 - \lambda^a_{1,s_0} + \lambda^a_{3,s_0s_1} - \lambda^a_{2,s_0s_1} = 0 \\
& x_0 \leq +x_3 \\
& x_3 \leq \lambda^a_{1,s_3} + 0.1\lambda^a_{2,s_3s_0} + 0.5\lambda^a_{2,s_3s_1} + 0.3\lambda^a_{2,s_3s_2} - 0.5\lambda^a_{3,s_3s_0} - 0.8\lambda^a_{3,s_3s_1} - 0.4\lambda^a_{3,s_3s_2} \\
& x_0 - \lambda^a_{1,s_3} + \lambda^a_{3,s_3s_0} - \lambda^a_{2,s_3s_0} = 0 \\
& x_1 - \lambda^a_{1,s_3} + \lambda^a_{3,s_3s_1} - \lambda^a_{2,s_3s_1} = 0 \\
& x_2 - \lambda^a_{1,s_3} + \lambda^a_{3,s_3s_2} - \lambda^a_{2,s_3s_2} = 0 \\
& x_3 \leq \lambda^b_{1,s_3} + 0.3\lambda^b_{2,s_3s_0} + 0.4\lambda^b_{2,s_3s_1} - 0.7\lambda^b_{3,s_3s_0} - 0.6\lambda^b_{3,s_3s_1} \\
& x_2 - \lambda^b_{1,s_3} + \lambda^b_{3,s_3s_0} - \lambda^b_{2,s_3s_0} = 0 \\
& x_3 - \lambda^b_{1,s_3} + \lambda^b_{3,s_3s_1} - \lambda^b_{2,s_3s_1} = 0
\end{aligned}$$

Free: $\lambda^a_{1,s_0}, \lambda^a_{1,s_3}, \lambda^b_{1,s_3}$

D Dining Philosophers

We analyze the classical Dining Philosopher Problem [$D.1$]. Briefly, n philosophers are sitting at a table with n available forks. Each philosopher can either think or eat: when he becomes *hungry*, he needs to pick both the fork on his right and on his left before *eating*. Since there are not enough forks to allow all philosophers to eat together, they need to follow steps according to a stochastic protocol to eat in turns. We consider this case study relevant because it can be used to model real shared-resources stochastic protocols [$D.1$], and because the size of the model n can be easily scaled to benchmark the time complexity of our routines.

We model the uncertainty of the philosophers in deciding which fork to pick first: while the nominal protocol assigns $0.5 - 0.5$ probability to the left and right fork, we assume that these values are only known with $\pm 10\%$ confidence. The parameters for each model of uncertainty corresponding to this level of confidence can be set using the approach suggested in [11]. For example, for the Interval model, the probabilities lie in the interval $[45\% - 55\%]$. Within this setting, we aim to determine which is the *quantitative* minimum probability for any philosopher to eat within k steps of the protocol after he becomes *hungry*. In PCTL syntax:

$$\mathbf{P}^{min}\left[\psi\right] := \mathbf{P}^{min}\left[\mathbf{F}^{\leq k}\{Eating\}\right] \tag{56}$$

with initial states $S_0 = Sat(hungry)$. Figure 2 shows the evolution of the probability of Equation (56) as a function of the number of protocol steps k. As expected, the probability of eating steadily increases as the number of steps increases. However, the plot also shows that adding uncertainty *decreases* this probability with respect to the nominal scenario (if no uncertainty is added, our results match those in [$D.1$]. The inset of Figure 2 shows the relative deviation in probability with respect to the nominal case: a $\pm 10\%$ uncertainty can cause a deviation up to 35% in the computed probabilities, and the deviation is always higher than 10% for $k \leq 60$. Further, the deviation is larger for the Interval and Ellipsoidal models, since they are the most conservative among the considered ones, as explained in Section 2.2 and Appendix A.

Lastly, we evaluate the runtime performance of our routines while varying the size n of the problem. We report three data points for each uncertainty model, corresponding to $n = 3, 4, 5$. Figure 3 shows that runtime increases approximately linearly with the number of states N. The discrepancy with the expected quadratic behavior can be explained considering that in this case study (and in most practical ones) not all actions $a \in A$ are available at each

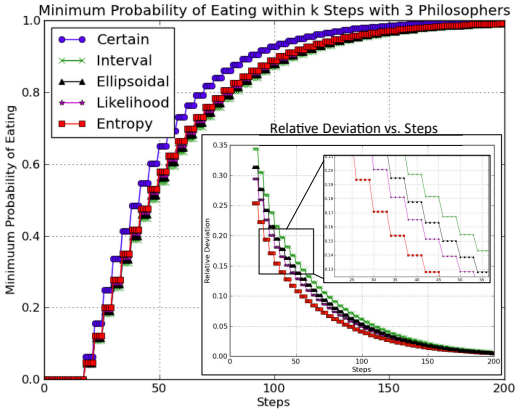

Fig. 2: Evolution of Equation (56) for increasing k and $n = 3$ philosophers.

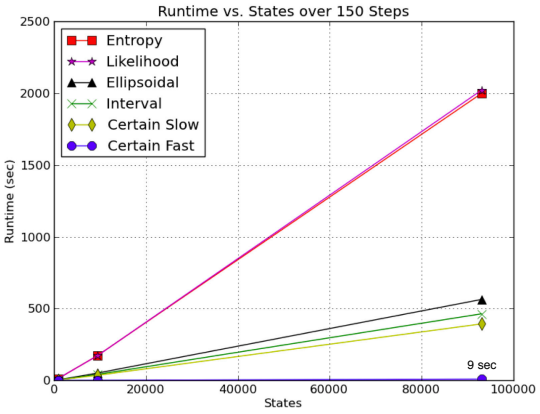

Fig. 3: Runtime vs. Number of States with $k = 150$ steps.

state $s \in S$ and the transition matrix $F^a \in \mathcal{F}^a$ is sparse. The interval and ellipsoidal models run faster because the inner convex optimization problems can be solved using simpler atomic operations (sum and multiplication) than the likelihood model (logarithm). Further, the routine for the interval model runs only 1.2× slower than the *Certain Slow* routine, and the penalty rises to 20× with respect to the *Certain Fast* routine: this result can be interpreted as the cost of not being able to use optimized library procedures for matrix-vector multiplication when adding uncertainty to the model. These results support our claim of good scalability of the proposed approach with respect to the model size.

[D.1] PRISM Model Checker - Dining Philosopher Case Study
- http://www.prismmodelchecker.org/casestudies/phil.php